EFFICIENT GENERATION OF SOCIAL NETWORK DATA FROM COMPUTER-MEDIATED COMMUNICATION LOGS

THESIS

Jason Wei Sung Yee, Second Lieutenant, USAF

AFIT/GCS/ENG/05-19

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GCS/ENG/05-19

EFFICIENT GENERATION OF SOCIAL NETWORK DATA FROM COMPUTER-MEDIATED COMMUNICATION LOGS

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science

Jason Wei Sung Yee, B.S.C.S.

Second Lieutenant, USAF

March 2005

AFIT/GCS/ENG/05-19

EFFICIENT GENERATION OF SOCIAL NETWORK DATA FROM COMPUTER-MEDIATED COMMUNICATION LOGS

Jason Wei Sung Yee, B.S.C.S.

Second Lieutenant, USAF

Approved:

/signed/	11 Mar 2005
Dr. Robert Mills (Chairman)	date
/signed/	11 Mar 2005
Lt Col. Summer E. Bartczak PhD (Member)	date
/signed/	11 Mar 2005
Dr. Gilbert L. Peterson (Member)	date

AFIT/GCS/ENG/05-19

Abstract

The insider threat poses a significant risk to any network or information system. A general definition of the insider threat is an authorized user performing unauthorized actions, a broad definition with no specifications on severity or action. While limited research has been able to classify and detect insider threats, it is generally understood that insider attacks are planned, and that there is a time period in which the organization's leadership can intervene and prevent the attack. Previous studies have shown that the person's behavior will generally change, and it is possible that social network analysis could be used to observe those changes.

Unfortunately, generation of social network data can be a time consuming and manually intensive process. This research discusses the automatic generation of such data from computer mediated communication records. Using the tools developed in this research, raw social network data can be gathered from communication logs quickly and cheaply. Ideas on further analysis of this data for insider threat mitigation are then presented.

Acknowledgements

I would like to extend my deepest gratitude to those who helped me complete this thesis. First, I would like to thank Dr. Mills for his mentorship and guidance throughout my time in AFIT. He was always knowledgeable, helpful, and understanding; I wouldn't be writing this without him. I would also like to thank Rick Calmes of SC, who took time out of his busy schedule to provide critical data to this research, and Tim Lacey of the LISSARD Lab, who made sure that our research machines ran the way they should. Thanks also to Dr. Lamont, Maj. Young, and Dr. Potoczny, who helped me mature as a computer scientist and problem solver while teaching me the coolest stuff I've ever learned.

I learned what I know about the Air Force from my fellow students, who never hesitated to share their friendship, experience, and sometimes food. Big thanks to Maj. Tobin, Maj. Zeitz, and Maj. Meadows who are incredible mentors and role models for a young lieutenant. Heartfelt thanks to Clyde, who helped me grow as an officer, a gentleman, and chef; I hope that I'll be able to make you proud and serve the Air Force as well as you. Thanks aplenty to my apartmentmates and best buddies Mike and Gary for keeping me sane and always entertained –good luck with your theses– G-G-G-G-UNT! Extra special thanks also to Matt & Sarah, Trasi, Ted, Theresa, Pat, Jason, Chris, the NSA kids, and the rest of the GCS CGO crew: these friends made graduate school even more enjoyable for me :)

Last but not least, uncountably infinite thanks and love to my family, who always loved, supported, and believed in me.

Jason Wei Sung Yee

Table of Contents

Page

Abstract iv

Acknowledgements v

List of Figures ix

List of Tables xi

List of Abbreviations xii

I. Introduction 1
- 1.1 Computer Security 1
- 1.2 The Insider Threat 2
- 1.3 Social Network Analysis 3
- 1.4 Purpose and Scope 3
- 1.5 Document Overview 4

II. Background on The Insider Threat and Social Network Analysis . 5
- 2.1 The Insider Threat 5
  - 2.1.1 A Definition of the Insider Threat 5
  - 2.1.2 Insider Threat Risk 6
  - 2.1.3 Countermeasures to the Insider Threat 9
  - 2.1.4 Characterizing the Insider Threat 13
- 2.2 Social Network Analysis 17
  - 2.2.1 Social Network Analysis Capabilities 21
  - 2.2.2 Difficulty in Gathering Social Network Data . . 26
- 2.3 Summary 27

III. System Implementation 29
- 3.1 Overview 29
  - 3.1.1 System Overview 29
  - 3.1.2 Expected Usage 31
  - 3.1.3 Privacy Protection 31
- 3.2 Gather SMTP and ProxyList Data 32
  - 3.2.1 Input 33
  - 3.2.2 Output 33
  - 3.2.3 Proxy List 33
- 3.3 ProxyListToUID Component 33

Page

3.3.1 Usage 34
3.3.2 Input 34
3.3.3 Output 35
3.4 SMTPLogSanitizer Component 36
3.4.1 Usage 37
3.4.2 Input 37
3.4.3 Output 37
3.4.4 Implementation Notes 38
3.5 SMTPLogParser Component 39
3.5.1 Usage 39
3.5.2 Input 39
3.5.3 Output 40
3.5.4 Implementation Notes 41
3.6 Database Functions Component 41
3.6.1 Database Setup 41
3.6.2 Importing Data into Message Table 42
3.6.3 Creating Sociograms 42
3.7 Summary 45

IV. Experiment Methodology and Results 47
4.1 Experiment Methodology 47
4.1.1 Goals and Hypothesis 47
4.1.2 Evaulation Metrics 47
4.1.3 System Boundaries 48
4.1.4 System Setup 49
4.1.5 Workload 49
4.1.6 Evaluation Technique 51
4.1.7 Summary of Experiment Methodology 52
4.2 Results and Findings 52
4.2.1 System Timing Test Results 52
4.2.2 Data Usefulness Test Results 56
4.3 Summary 57

V. Conclusion and Recommendations 62
5.1 Research Objectives 62
5.2 Impact 62
5.2.1 Immediate Impact 62
5.2.2 Long-Term Impact 62
5.3 Future Goals 63

5.3.1 Characterizing Behavior 64
5.3.2 Gathering More Live Data 64
5.3.3 Determining the Best Parameters 64
5.3.4 Expansion to Other CMC Records 65
5.3.5 Tool Improvements 65
5.4 Summary 66

Bibliography 67

Vita 70

Index 70

List of Figures

Figure		Page
1.	Dollar Amount of Losses by Type [27]	7
2.	Types of Attacks or Misuse Detected in 2004 (by percent) [27]	8
3.	Reporting Cyberattacks in 2004 [27]	14
4.	Sociogram of AFIT, from August 10-15 drawn by MAGE [26]	17
5.	Sociogram data in graphical form [17]	21
6.	Social Network Data from Antony and Cleopatra Gathered by PieSpy [22]	24
7.	Krebs' Mapping of the 2001 Terrorist Network [16]	28
8.	System Process and Components	30
9.	System Implementation	31
10.	File Permission to Protect Privacy	32
11.	Proxy List File Format	33
12.	ProxyListToUID Component	34
13.	Format of UID list	35
14.	Terminal Output for ProxyListToUID	35
15.	SMTPLogSanitizer Component	36
16.	Terminal Output for SMTPLogSanitizer	38
17.	SMTPLogParser Component	39
18.	Sample Sanitized Log Input for SMTPLogParser	40
19.	SMTPLogParser Output for the Sample in Figure 18	40
20.	Terminal Output for SMTPLogParser	41
21.	Database FunctionsComponent	42
22.	SQL Query to Create Email Message Table	43
23.	SQL Query to Import Data from SMTPLogParser	43
24.	Exported Query Results	44

Figure		Page

25. Header for Top of Exported Query Results 44
26. UCINet Import Error 45
27. Header for Top of Exported Query Results with 1491 Nodes . . 45
28. SQL Query to Create a Sociogram 46
29. SQL Query to Create an Internal Sociogram 46
30. SQL Query to Create an Internal Sociogram Limited by Date (September 2004) 46
31. SQL Query to Create an Internal Sociogram Limited by Date (September 2004) and Number of Recipients (less than 20) . . 46
32. System Under Test 48
33. Bonacich Power Metric 58
34. UCINet Freeman Degree Metric 58
35. UCINet Output for *k*-core Metric 58
36. NetDraw Visualization of Social Network Data for December 2004 59
37. Egonet of Actor with UID 59 59
38. Subset of the December 2004 Sociogram 60
39. Spring-Embedding Visualization 61
40. Gower Visualization 61
41. Circular Visualization 61
42. Multi-Dimensional Scaling Visualization 61
43. Future Targets for Using SNA to Mitigate the Insider Threat . 63

List of Tables

Table		Page
1.	Example of Standard Sociometric Data	19
2.	Example of an Undirected, Binary Sociogram, Social Network Data of an Organization [17]	20
3.	Example of a Directed, Weighted Sociogram, Social Network Data of an Organization [17]	20
4.	Data Size	49
5.	Component Runtimes for Three Months	53
6.	Component Runtimes for One Month	53
7.	ProxyListToUID Runtimes in Milliseconds	54
8.	SMTPLogSanitizer Runtimes in Minutes	54
9.	UIDs Added by the SMTPLogSanitizer	54
10.	SMTPLogParser Runtimes in Minutes	55
11.	Number of Emails and Ties	55
12.	Database Creation and Import Times in Seconds	56
13.	Database Query Times in Seconds	56

List of Abbreviations

Abbreviation		Page
IDS	Intrusion Detection System	1
COTS	Commercial Off The Shelf	2
SNA	Social Network Analysis	3
CMC	Computer Mediated Communication	3
CSI	Computer Security Institute	6
CDIS	Computer Defense Immune System	11
AIS	Artificial Immune System	11
SS	U.S. Secret Service	14
ONA	Organizational Network Analysis	22
IRC	Internet Relay Chat	23
SMTP	Simple Mail Transfer Protocol	29
UID	Unique Identification Number	31
SMTP	Simple Mail Transfer Protocol	47
NCSA	National Center for Supercomputing Applications	49

EFFICIENT GENERATION OF SOCIAL NETWORK DATA FROM COMPUTER-MEDIATED COMMUNICATION LOGS

I. Introduction

1.1 Computer Security

Computers and networks have revolutionized the methods and speed in which work is done. Ever-increasing amounts of information can be collected, stored, processed, and disseminated in shorter amounts of time and at reduced cost. Computers and networks have been critical to an organization's ability to function. Defenseof these information systems is therefore essential.

The threat of attacks on computer systems has been recognized at the highest levels. In 1998, President Clinton addressed the potential danger of cyberattacks on the computer systems that are necessary for the nation's critical infrastructure [23]. Presidential Decision Directive 63 (PDD-63) states that it is the President's intent that the "United States will take all necessary measures to swiftly eliminate any significant vulnerability to both physical and cyber attacks on our critical infrastructures, including especially our cyber systems." This kind of unconventional attack may come from foreign governments, foreign and domestic terrorist organizations, and foreign and domestic criminal organizations.

Computer security in general has come a long way in defending against cyberattacks. The NSA prescribes a defense-in-depth methodology consisting of discrete layers of defense using routers, firewalls, and intrusion detection systems (IDS) to defend a computer network [8]. These layers protect individual machines, machines subject to access controls, and defended by anti-virus programs. In terms of defending against entities outside of the system without access, the defense-in-depth methodology protects critical information relatively well. There are many viable commercial-

off-the-shelf (COTS) and open-source firewalls, IDS, and anti-virus programs that can be readily integrated into a defense-in-depth security policy. Moreover, Microsoft, the industry leader in operating system software, has made security a cornerstone of their current and future software with the Trustworthy Computing initiative [20].

1.2 The Insider Threat

No matter how fortified a network, or how strong and complete a security policy, no system is protected from abuse by authorized users. Seemingly harmless mistakes or an unforeseen chain of events caused by careless or uninformed users can lead to a dangerous security breach. Or even worse, a renegade administrator can wreak havoc upon a system. While it is necessary to invest time and money to fortify networks against external threats, it is also necessary to assess the threat from within.

The insider threat is well-documented, and many view it as the greatest risk and be the most difficult to detect. For the purposes of this research, "insider" refers to an individual who has been granted trusted access to an organization's information resources, including the computing network and data stores. The insider threat then, is the threat that an insider would use their privileges to attack or otherwise misuse the organizations information resources.

The insider threat is an increased danger to the system because an insider has more knowledge of the system and its protection than a typical outsider. Moreover, since the insider is already a trusted entity in the system, malicious acts may be within established privileges and might not be detected. Given a large variety of insider attacks, the difficulty of characterizing malicious insiders, and relative lack of data characterizing, research in detecting insider threats has been limited. The best method of mitigating the insider threat is to constantly monitor user behavior and have managers intervene when appropriate. Unfortunately, managers can only monitor subordinate behavior so much, and it is a difficult task where there are large numbers of subordinates and given current business practices.

1.3 Social Network Analysis

Social network analysis (SNA) is a relatively new field of psychology and social behavior founded upon the idea that the relationships between people are just as important as the attributes of people. Social network analysis provides a rigorous method of analyzing the interactions between members of a group. Since social network data can be analyzed with graph theory concepts, the speed and power of computers can be leveraged. Thus, social network analysis is an increasingly powerful tool for quickly analyzing relationships of individuals. With broad application, social network analysis has been used to help streamline business processes, improve internal organizational communication, and even position routers in a network topology with great success.

In the context of mitigating the insider threat, it is possible that social network analysis can identify individuals who may be higher risks to the organization. Studying user behavior based on social network analysis metrics could potentially provide means of characterizing the insider threat. Changes in social network analysis metrics may indicate a change in an actor's behavior. As Shaw writes, a change in behavior may be indicative of a potential insider threat [30].

Unfortunately, building a social network is time consuming and may require a degree of cooperation from the subjects being studied. Additionally, a social network will also change over time, based on shifts in workload or project prioritization, which might render existing maps obsolete. The purpose of this research then is to investigate means of building a social network map using automated tools/proceduresand administrative logs of computer mediated communications (CMC) (e-mail, telephone logs, instant messages, etc.) as a source of information.

1.4 Purpose and Scope

The purpose of this research is to evaluate the timeliness of execution and data usability of a system that automates building social network data from electronic mail

logs. It is the intent of this research to prove that a system that generates useful social network data in a reasonable amount of time can be created. This system can then provide raw social network data to other tools that use social network analysis, and aid in the mitigation of the insider threat. This research is not intended to provide a detector of tendencies of potential insider threats but instead a form of middleware to process raw data into a form suitable for follow-on analysis. The system created in this research can be used to provide social network data to help train a tendency detector in the future.

1.5 Document Overview

This thesis is organized as follows. Chapter 2 contains an overview and review of insider threat and social network analysis concepts. The methodology to and experimental procedure of this research is described in Chapter 3. Chapter 4 discusses the system created and is something of a user's manual for it. Chapter 5 reports test results showing that the system is viable based on the methodology described in Chapter 3. The conclusions and impact, along with a vision for constructing an insider threat tendency detector and recommended areas for future research are discussed in Chapter 6.

II. Background on The Insider Threat and Social Network Analysis

This first section of this chapter describes the insider threat, the significant risk it poses, current tools and research efforts toward mitigating its risk, and how to characterize malicious insiders. The second section provides a definition for social network analysis, its capabilities, and the difficulty of gathering social network data.

2.1 The Insider Threat

2.1.1 A Definition of the Insider Threat.

In its proceedings about mitigating the insider threat, RAND provides a straightforward definition: the insider threat is the threat of "Authorized users performing unauthorized tasks" [24]. This is an extremely broad definition with no restrictions or specifics on severity, intention, perpetrator, or what constitutes an unauthorized task. For the purposes of this research, "insider" refers to an individual who has trusted access to an organization's information resources, to include the computing network and data stores. The insider threat then, is the threat that an insider would use his privileges to attack or otherwise misuse the organizations information resources. The insider threat is an increased danger to the system because an insider has more knowledge of the system and would be able to use this information to be more harmful. Moreover, since the insider is already a trusted entity in the system, the insider's malicious acts may fall within normal privileges and thus this behavior might not be observed or detected as they fall within normal behavior.

A straightforward example of an insider is a system administrator who knows he will be laid off within the year. The day before he is laid off, he changes the root password of an important database and tells the company that they can have the password if they pay him a large sum of money. It could take months to crack the password, and while that data is encrypted, the police would have a hard time prosecuting the system administrator if he claims to have "forgotten" the password. Furthermore, the system administrator may leak this information to a newspaper and

cause bad press for the company. This places the company in a tight bind in which the company may ultimately pay up to avoid the bad press and legal costs.

Espionage is possibly the most serious form of the insider threat. The case of former FBI agent Robert Hanssen, who was convicted for spying for Russia, is an extreme example of how insiders can take advantage of their access and authorizations. Over a span of more than 15 years, Hanssen provided his Russian contacts with highly classified documents and details about U.S. intelligence sources and electronic surveillance taken directly from his employer, the FBI. Because Hanssen was an authorized user, his activities didn't raise any suspicion. While Hanssen used a variety of devices for stealing data, he also repeatedly walked out of his FBI office carrying classified paper documents in his briefcase, which in turn, he handed over to his Russian contacts. Hanssen manually and electronically stole information from the FBI for his own financial gain, and he did it for more than 15 years without trouble because he was a trusted insider [5].

The insider threat can manifest itself in various forms, making it extremly difficult to detect. Also, with the insiders' additional access and knowledge, the damage that can be done from insider activity can be extremely severe.

2.1.2 Insider Threat Risk.

As can be imagined, the damage from insider attacks can be more devastating than other attacks. The insider threat is an increased danger to the system because an insider often has knowledge of the system's vulnerabilities and would be able to use this information to do more damage. Likewise, an insider knows what information is most valuable to the organization. At the same time, the insider threat risk is elevated because a high degree of technical skill is not required to carry out an insider attack.

Figure 1 shows estimates of the losses caused by type of computer security incident as reported in a survey taken by the Computer Security Institute (CSI) and FBI in 2004 [27]. By definition, sabotage and insider net abuse are insider attacks. In

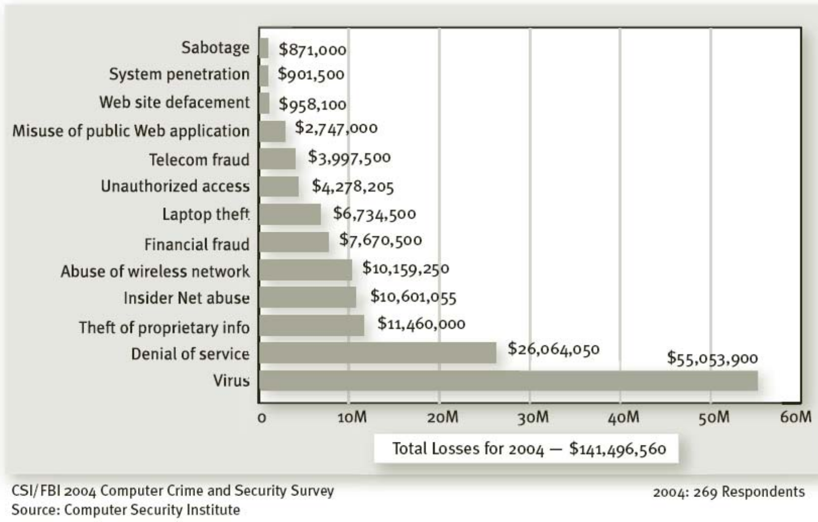

Figure 1: Dollar Amount of Losses by Type [27]

addition, theft of proprietary information and financial fraud often require insiders, and unauthorized access is more costly when done by insiders. With this in mind, the insider threat accounted for over $30 million in damages [25].

The risk of losing proprietary information is especially dire when dealing with classified information, information that can endanger national security. There is a lot of classified information like this; there is an extremely high potential for grievous damage, especially when insiders likely know what classified information carries the greatest value on the black market [13]. The insider is often indispensable to "outside" groups (including foreign interests) as they can provide help and information [30].

A study by PricewaterhouseCoopers, the U.S. Chamber of Commerce and the American Society for Industrial Security International estimated that companies lost up to $59 billion in intellectual property and proprietary information between July 2000 and June 2001. The largest average dollar value of loss per incident occurred in research and development ($404,000), followed by financial data ($356,000) [36].

In the CSI/FBI survey as shown in Figure 2, incidents insider abuse of net access were reported by 59% of corporations, theft of proprietary information by 10%, and sabotage by about 2% [27]. While the percentage of incidents reported of these likely

insider attacks decreased sharply from 2003, the insider threat is still very pervasive and is second only to the ubiquitous virus threat. This decrease corresponds to a general decrease in all instances of attacks, likely the result of increased awareness and importance of information security. A similar decrease occurred in 2002, yet reports of incidents rose in 2003.

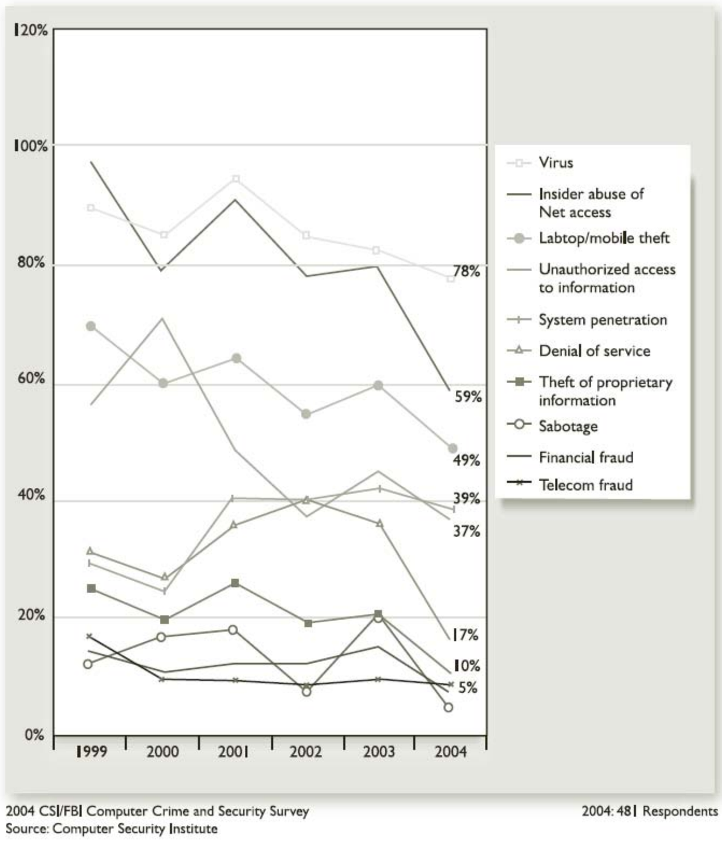

Figure 2: Types of Attacks or Misuse Detected in 2004 (by percent) [27]

The Secret Service reports that risk associated with the insider threat is even higher as attacks often require no technological expertise greater than what is required to do the job. It was found that 87% of insider attacks in the banking and finance sector required little or no technical expertise [25]. In fact, 43% of those attacks were carried out while the perpetrator was logged in under their own credentials and only 23% of the malicious insiders held technical positions in information technology.

It is clear that the potential for damage by insider attack is severe. With their knowledge of the system, insiders can focus on vulnerabilities and high-value targets without needing a lot of technical expertise. Unfortunately, strategies and countermeasures to combat the insider threat are limited and none are truly effective.

2.1.3 Countermeasures to the Insider Threat.

Current techniques used in computer security are not able to cope with the dynamic and increasingly complex nature of computer systems and their security [2]. Since there are many different forms that insider attacks can take, the difficulty these countermeasures face in detect insider attacks is tremendous. Thus, to further mitigate the insider threat, something must be done on the preventative side, looking for behaviors in insiders that indicate the potential to become insider threats.

The best countermeasures against insider attacks are general management strategies, different types of intrusion detection systems, and honeypots. While all of the aforementioned strategies have their strengths and should be used in conjunction to mitigate the insider threat, they all have shortcomings. The underlying theme is that it is extremely difficult to counter the insider threat by attempting to detect it with current technologies.

2.1.3.1 General Management Strategies.

At the most fundamental level, security is a people problem that requires people or people-based solutions to be solved. Since it is impossible to hire only people that can be completely trusted it is imperative that there are good managers and managerial strategies that foster good security practices. Shaw writes that management that keeps its employees satisfied and loyal prevents insider incidents [30]. In a report on the insider threat to US Government Information Systems in 1999, five strategies were suggested in order of relative importance:

- Create good access controls
- Keep access logs
- Audit the actions of individuals

- Protect sensitive information
- Detect covert access [12]

The first three of these strategies, if implemented extremely rigorously, can eliminate the majority of insider threats. Ensuring good access controls means setting an access controls that adhere to the principle of least privilege. This means that a user should only have the privileges required to do their duties. This way, malicious insiders can only compromise information that they have access to. Keeping access logs and reviewing the actions of individuals provides evidence and accountability for user actions. Additionally it also deters users from attempting to subvert security as they know that they are being watched. Unfortunately, it is not possible or feasible to audit every action taken by every user, and even if it were possible, it would be difficult to determine the intentions of the actions [28].

Preventing and detecting covert access to sensitive information and networks is much easier to say than do, as the actions are covert by nature. A strong defense in depth security policy including firewalls and good access controls are essentially the best way to prevent covert accesses. Detecting covert access implies the usage of intrusion detection systems (IDS). Since the publishing of this paper, these policies have become standard at any organization [27]. Still, insider attacks continue to be reported frequently.

2.1.3.2 Intrusion Detection Systems.

As their name implies, intrusion detection systems help detect and identify intrusions and attacks on a computer system or network. To detect intrusions, an IDS analyzes and monitors information like network traffic, access logs, and host data. An IDS is categorized by what it protects: a host-based IDS focuses primarily on host data and a network-based IDS analyzes network traffic. An IDS can also be categorized based on its approach for detecting an attack. The main categories are signature-based, anomaly-based, and compound or hybrid.

A signature-based IDS like SNORT is effective in detecting an insider attack if the insider uses known attacks on the system and the IDS is properly configured to monitor internal traffic [31]. These systems resemble anti-virus systems as they look for patterns in packets and packet data, whereas anti-virus systems search for patterns in file contents. Like an anti-virus system, a signature-based IDS requires updates for new rules just as anti-virus systems require updates of their virus definitions. Rules are based on individual attacks and require regularity to be effective. Effective rules can be written because many network attacks use the same tools and have similarities. A signature-based IDS is effective because many cyberattacks are carried out by "script kiddies," novice-level cybercriminals, who use the same tools. However, a signature-based IDS does nothing if an attack's signature is not in the rule set of the IDS or the data is not abnormal. Insider attacks have so many different forms, instances, and contexts that it is difficult to find patterns for them all. Moreover, an insider exploit may be done with actions expected of the insider.

AFIT's Computer Defense Immune System (CDIS) is an advanced network-based IDS that is somewhat effective system in mitigating the insider threat. The CDIS is a hybrid IDS based upon the Artificial Immune System (AIS), which is modeled after the antibodies in the human immune system [2]. When the human immune system detects a foreign antigen, something that is not a part of the human "self," it creates antibodies to destroy those specific antigens. Similarly, when the CDIS detects foreign sequence actions, it creates a rule that triggers a future response to that sequence or sequences like it. This approach uses evolutionary algorithms to create rules that detect intrusions similar to the ones detected. However, as a hybrid IDS, it suffers from the same problems that plague an anomaly-based IDS, primarily the need to train the IDS with "normal" data. Since insider attacks can consist of the same commands that a regular user would use on a normal basis, an anomaly-based IDS will most likely be unable to detect the attack.

An IDS is an essential part of the defense in depth strategy and should always be deployed. While an IDS may not be able to detect many insider threats, it is still very effective in detecting other attacks.

2.1.3.3 Honeypots.

Spitzner writes about how honeypots are used to detect and study insiders [32]. A general definition for a honeypot is "an information system resource whose value lies in unauthorized or illicit use of that resource." Generally, a honeypot is a program that emulates a server that provides an exploitable service or hosts important-looking information. Honeypots can be used for slowing down or stopping automated attacks, capturing new exploits to gathering intelligence on emerging threats or early warning and prediction.

An example of a COTS honeypot is the Symantec Decoy Server (SDS) [33]. One thing the SDS does is pretend to be a file server on the network containing sensitive information. Since this server is not supposed to exist, any user trying to access it is already doing something wrong.

Honeypots are great for studying attackers, detecting attacks on fake services, and deterring potential insiders. They should be implemented as a part of any defense in depth strategy. However, honeypots provide no warning or protection for actual targets and the malicious insider often knows exactly what targets to exploit.

2.1.3.4 Countermeasures Summary.

The previously mentioned countermeasures should all be implemented as part of a defense in depth strategy. Yet, all of the protection provided is still not enough to significantly mitigate the insider threat. Given the many shapes and sizes that insider attacks can have, the inability to detect attacks as they happen seems logical. Thus, to further mitigate the insider threat, the most logical path is to pursue different preventative measures.

The most effective way to prevent an insider attack may be the most fundamental. At the people level, managers and leaders must observe their team members' behaviors well and give them extra attention when they need it; it is hypothesized that it is possible to prevent insider attacks from happening with simple human intervention [28]. Unfortunately, getting managers to know their teams well is sometimes difficult.

In plain terms, it is expected that a manager can characterize the behavior of each individual subordinate, be able to detect a change in an individual, monitor behavior, intervene if necessary. The problem is that managers don't often know their teams well, not because they are bad managers, but because it is difficult to do so in today's work environments. Managers often supervise teams that are separated geographically and the problem of knowing teammates is compounded because many technical workers are introverts [30].

In all, current countermeasures are hindered by the lack of characteristics or tendencies shared insiders. If this knowledge is known, managers can look for these characteristics in their subordinates and take the necessary action.

2.1.4 Characterizing the Insider Threat.

This section discusses different classifications of insider attacks and insiders, and explain why characterization and targeting malicious insiders with preventative measures are good strategies for mitigating the insider threat. Characterizing the insider threat means learning who the insider is, what the attack is, and why the insider attacks.

Characterizing the insider threat is difficult because there is limited data and literature on insider attacks. There is limited information because many companies do not report insider attacks in order to minimize negative attention and monetary loss. Moreover, the company would just invite more attacks on systems that appear to be vulnerable. CSI/FBI reports that more than half (Fig. 3) of companies did not report that they were subject to insider attacks. [27]

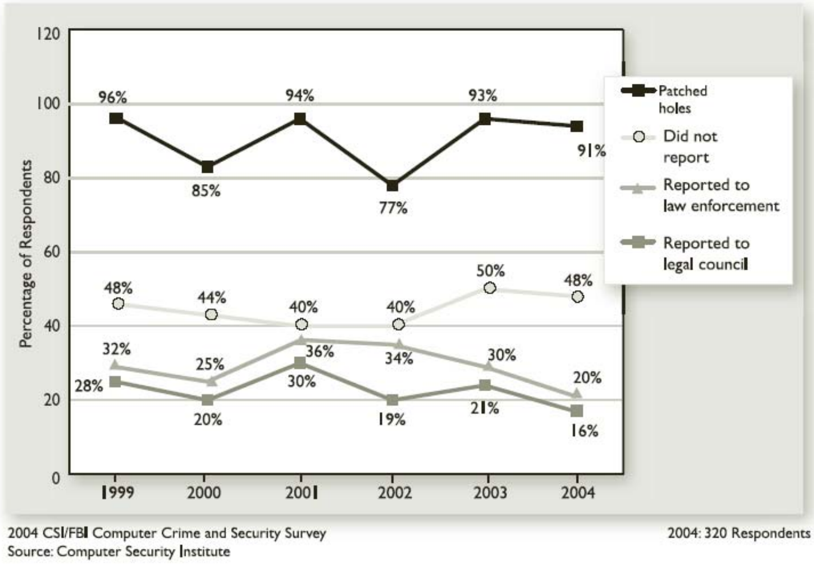

Figure 3: Reporting Cyberattacks in 2004 [27]

Thankfully, studies have been done with the data reported to the authorities. Along with the CSI/FBI yearly study, the U.S. Secret Service (SS) and the CERT Coordination Center did a case study on insider incidents in the banking and finance sector, and Shaw did extensive research on insiders.

2.1.4.1 SS/CERT Study.

The U.S. Secret Service and the CERT Coordination Center (SS/CERT) did a case study on insider incidents in the banking and finance sector, areas that have always placed a strong emphasis on security. However, these companies in these sectors were still the victims of malicious insider attacks. Thus, SS/CERT studied the incidents and published their findings [25].

In the study, SS/CERT found that the perpetrators did *not* share a common profile in terms of age, gender, marital status, employment position. They also found that few of the insiders were considered by management or co-workers as difficult to manage, untrustworthy, or were perceived to be disgruntled employees.

SS/CERT found that incidents were carried out by non-technical workers and required little or no technical sophistication. Very few cases even used scripts or

programs; a signature-based IDS would not detect these kinds of attack. In fact, most of the attacks were manually detected by people who were not responsible for security.

Central to this thesis, SS/CERT found that 81% of perpetrators planned their actions in advance. A planned incident can be stopped in the planning phase, whereas it is more difficult to stop a spur-of-the-moment crime. And of the cases documented, 39% of insiders had come to the attention of either a supervisor and/or coworker for some concerning behavior prior to the incident. Examples of these behaviors include increasing complaints to supervisors regarding salary dissatisfaction, increased cell phone use at the office, refusal to work with new supervisors, increased outbursts directed at coworkers, and isolation from coworkers. If there is any trend in the behavior of insiders, this is it: the perpetrators plan and exhibit changes in behavior. Thus, if it is possible to find changes in behavior during the planning phase, it is possible to prevent an insider attack.

2.1.4.2 Shaw's Studies.

Dr. Eric Shaw produced some seminal research on the insider threat. Though his primary focus is on the technically-skilled worker, his findings, especially those about the feasibility of preventing insider attacks, echo those from SS/CERT.

Shaw describes a specific kind of insider, the Critical Information Technology Insider (CITI) [30]. A CITI is an employee whose job deals directly with critical computer systems. These information technology specialists' job functions elevate them well above the average end-user in terms of skill, access and potential damage.

Shaw created a typology of eight categories that the motivations of the perpetrators fell into. Explorers, mostly harmless, are curious users who wander into the wrong areas in a network. Good Samaritans are willing to put aside rules to get the job done better. Hackers are motivated to show off and boost their self-esteem. Machiavellians use corporate systems to advance their careers. Avengers seek vengeance, career thieves use computer systems to make money, and moles are spies. Since these moti-

vations are broad and difficult to detect without human interpretation, it is difficult to classify insiders on motivation.

Shaw also identified six personal characteristics with direct implications for risk. They often have a history of personal and social frustrations, computer dependency, ethical "flexibility," reduced loyalty, a sense of entitlement, and a lack of empathy [29]. These characteristics are difficult to detect without a good personal knowledge of the subject. Moreover, no insider would openly admit to these characteristics. Again, good managerial practices and close interaction with employees is key to identification of these characteristics.

Shaw notes that acute situational stressors can trigger an emotional reaction leading to impaired judgment and reckless or vindictive behavior. Examples of acute situational stressors are marital or family problems, episodes of substance abuse, disappointments at work, threatened layoffs, or other stressful life events [30]. Shaw reports that if warning signs like acute situational stressors are detected, managers have a time frame in which to intervene and prevent an insider attack. While many cases first appear to involve disgruntled employees who execute a single massive attack after receiving dramatic news, perceiving a slight, or suffering a setback, closer examination of the case histories reveals that many employees demonstrated clear signs of disgruntlement and committed less serious violations leading up to the ultimate act [28]. This suggests that the most devastating attacks could have been prevented if the early indications of dissatisfaction had been recognized or taken more seriously. This information again places the burden upon the manager to be well-informed.

In many cases, supervisors were aware of insider attacks but did not appreciate their significance. There were also many cases in which the managers were unaware of their subordinates' changing behaviors. Shaw concludes that there is a strong need for supervisor training, improved communications with security, and independent channels by which security can monitor risk situations, other than via supervisor notification [29].

2.1.4.3 Mitigating the Insider Threat.

Given the difficulty of detecting the insider threat with the defense-in-depth methodology and the best available tools, the insider threat is best mitigated through good managerial practices in which supervisors are in close contact with their subordinates and have a good communication line with security. Managers and security personnel would be able to mitigate the insider threat more effectively if they had more useful information about their employees' behavior. This need for more information can potentially be met with social network analysis.

2.2 Social Network Analysis

"The truth lies within the social fabric that connects people to people and people to content." – Peter Morville [21]

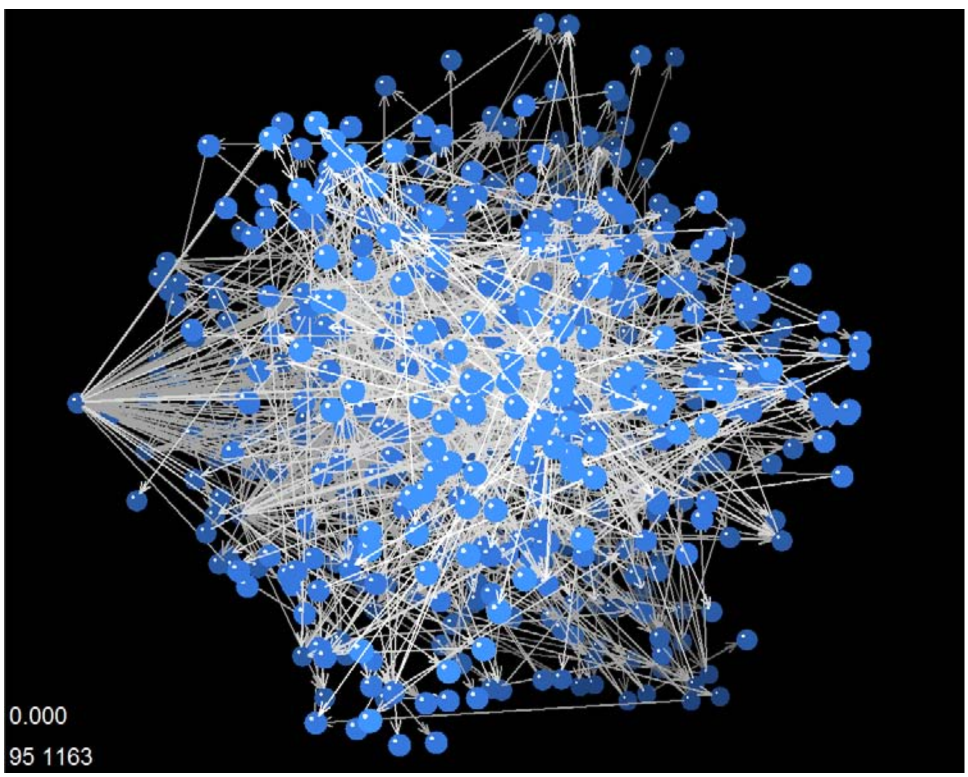

Figure 4: Sociogram of AFIT, from August 10-15 drawn by MAGE [26]

Social Network Analysis draws from the fields of Psychology and Sociology, to study people and the relationships between groups of people [35]. While it is generally

easier for a sociologist to study individuals and their attributes, the ultimate goal of a sociologist is to understand the society itself: relationships within a group of social entities and how they affect the individuals. This is the social network analysis perspective.

The roots of social network analysis come from the work of J.L. Moreno, who introduced the sociogram, a picture of the interpersonal structure of a group in 1934. Moreno, considered the father of social network analysis brought forth the approach called sociometry. Recognition that sociograms could be used to study social structure led to a rapid introduction of analytic techniques. Much of the social network analysis work that is known today came out interdisciplinary efforts thanks to the broad appeal of the social network perspective [35].

Social network analysis is an interdisciplinary behavioral science specialty. It is grounded in the simple, plainly obvious observation that social actors are *interdependent* and that the links among them have important consequences for every individual. In social network analysis, the relational ties between actors are primary and the attributes of actors are secondary.

Wasserman describes social network analysis as a "distinct research perspective within the social and behavioral sciences; distinct because social network analysis is based on an assumption of the importance of relationships among interacting units" [35]. Instead of focusing on the attributes of the individuals as is done in standard social analysis, social network analysis focuses on ties, the interactions and relationships between the individuals as a way of characterizing their behavior. When social network analysts study ties, they interpret their functioning in the light of the actors' relations with other network members [10].

2.2.0.4 Social Network Data.

For example, a standard sociological study on the importance of individuals in an organization might count the number of phone calls each individual makes and receives and take into account the attributes of the callers, like age and gender as

Name	Gender	Age	Calls
Andre	M	34	18
Beverly	F	32	17
Carol	F	49	9
Diane	F	19	32
Ed	M	54	10
Fernando	M	30	19
Garth	M	24	22
Heather	F	55	7
Ike	M	43	6
Jane	F	64	4

Table 1: Example of Standard Sociometric Data

shown in Table 1. The study then concludes by hypothesizing a relationship between the attributes and the measured importance of the sample.

On the other hand, a social network approach analyzes *who is calling whom* and the groups that are formed as a result. A secretary, for example, may make a lot of phone calls, but is not necessarily the most important person in the organization. The social network perspective looks at the relationships between the actors and then tries to unearth how actors are structured relative to each other. Simple network data is composed of actors, the entities being studied, and ties, representing a relationship between actors. Social network data is often displayed in an adjacency matrix as shown in Table 2.

This data is a sociogram or social network map and is the fundamental unit of study in social network analysis [9]. In Table 2, the presence of a 1 at (x, y) in the matrix represents the presence of a tie between the two actors x and y. Since there is a 1 where row C and column F intersect, actor C has a relationship with actor F. Notice that there is no mention of the attributes of the actors in this data. In this example, the data is binary: each tie is either 0 or 1. The data does not necessarily have to be this way as the strength of the tie can also be used. For example, actor F is connected to actor D with a tie strength or weight of 7 in Table 3. Also, not every tie in must be reciprocated (a tie between A and B is present if a tie between B and

	A	B	C	D	E	F	G	H	I	J
A	-	1	1	1	0	1	0	0	0	0
B	1	-	0	1	1	0	1	0	0	0
C	1	0	-	1	0	1	0	0	0	0
D	1	1	1	-	1	1	1	0	0	0
E	0	1	0	1	-	0	1	0	0	0
F	1	0	1	1	0	-	1	1	0	0
G	0	1	0	1	1	1	-	1	0	0
H	0	0	0	0	0	1	1	-	1	0
I	0	0	0	0	0	0	0	1	-	1
J	0	0	0	0	0	0	0	0	1	-

Table 2: Example of an Undirected, Binary Sociogram, Social Network Data of an Organization [17]

	A	B	C	D	E	F	G	H	I	J
A	-	3	1	1	0	1	0	0	0	0
B	1	-	0	1	4	0	1	0	0	0
C	5	0	-	1	0	4	0	0	0	0
D	1	2	1	-	1	1	1	0	0	0
E	0	1	0	1	-	0	1	0	0	0
F	1	0	1	7	0	-	1	1	0	0
G	0	9	0	1	1	2	-	1	0	0
H	0	0	0	0	0	1	1	-	3	0
I	0	0	0	0	0	0	0	1	-	1
J	0	0	0	0	0	0	0	0	2	-

Table 3: Example of a Directed, Weighted Sociogram, Social Network Data of an Organization [17]

A is present) as is shown in Table 2; it is not difficult to imagine if the relationship is directional, like “owes money to.”

From the matrix, it is possible to construct a visualization of the actors’ relationships with each other by making each actor a node in a graph and drawing an edge between them if a relation exists. This visualization can show the structure of relationships as shown in Figure 5.

As they are intrinsically graphs, sociograms adhere to the rules of graph theory. Graph theory is very useful in social network analysis because it provides a vocabulary

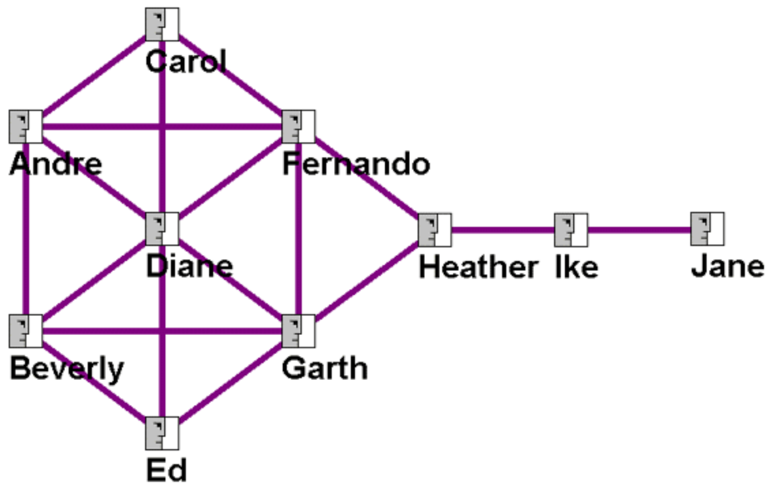

Figure 5: Sociogram data in graphical form [17]

to label and denote social structural properties. It also gives social network analysts the tools used in studying graph theory: mathematical operations and concepts that quantify and measure structural properties. Perhaps most importantly, giving social networks a rigorous, symbolic representation means that the power of computers can be leveraged to aid in research.

2.2.1 Social Network Analysis Capabilities.

Before the widespread availability of computers, the fruits of labor of a lengthy interview or observational study were tedious arithmetic to calculate metrics of the social network data and analyze them. Even more work is required to create a visualization for the social structure. Computers have changed the speed of social network analysis dramatically.

Social network analysis programs like UCINet [4], Pajek [3], and KrackPlot [14] can calculate the properties of a graph with a few clicks of a mouse [10]. These programs use graph theory algorithms and concepts to quickly calculate social network metrics.

The primary goal of social network analysis is to understand the actors based on their relationships. Understanding the actors means gaining information about

the tendencies of the actors and being able to predict the attributes of the actors as well as the actions the actors would take if ties or actors were to disappear. The basic metrics of social network analysis revolve around activity, betweenness, and closeness. Activity is often measured in terms of degrees, the number of ties that an actor has. Betweenness is a measure of how many shortest paths between two actors go through a specific actor. Closeness is a measure of how few connections are required to connect to other actors. These basic social network concepts are used to calculate almost all of the social network metrics and gain information about the entire network as well as the actors themselves.

With the right parameters and usage, the metrics of social network analysis may one day be used to help mitigate the insider threat. Since these metrics correspond to the behavior of users, a change in these measurable metrics indicate a change in user behavior. If these changes in behavior are indicative of potential insider behavior, the user's supervisor can be notified and an insider attack may be averted. Currently, many different applications and uses of analysis on social network data have been proposed and implemented. Some intriguing and effective uses were to improve the productivity of organizations, optimizing the use of routers, and tracking terrorists.

2.2.1.1 Organizational Network Analysis.

Rob Cross of the University of Virginia uses social network analysis to analyze organizational networks and help improve company productivity by increasing collaboration and information flow [7]. Fundamental to Organizational Network Analysis (ONA) is the idea that people work well when they work together, and even better when the right people are connected. Social network analysis find bottlenecks of information flow, organizational hermits who are not collaborating, and elitist groups that don't interact with those outside of the group. With the information he acquires, Cross recommends ways to restructure or join groups through meetings or the hiring of mediators. Among other things, ONA has been used to help integrate newly

merged companies, improve strategic decision making in top leadership, and promote creative thought.

Given its results, ONA is very effective, as some of Cross' most prominent clients are: American Express, Accenture, Asea Brown Boveri (ABB), Abbey National, A D Little, Aventis, Bank of Montreal, BP, Bristol-Myers Squibb, Capital One, Cardinal Healthcare, Conoco, CSC, Eli Lilly, EnCana, FAA, Halliburton, IBM, Intel, Mars, Martha Jefferson Hospital, McKinsey, Microsoft, Nortel, Novartis, NSA, PriceWaterhouse [7].

2.2.1.2 The Social Life of Routers.

Vladis Krebs showed that social network analysis ideas can help routers optimize their connections in a network topology [15]. Optimal connections in a network topology mean that the hop count (number of intermediate steps required to connect two routers) between all routers is minimized. This has excellent application, especially for cable companies with which there is a large cost associated with laying cable and connecting different nodes. Since the idea of a relational tie in a social network is analogous to a connection between two routers in a computer network and hop counts are calculated the same way that geodesic distance is calculated, Krebs turns the topology of a computer network into a sociogram and takes some social network analysis metrics.

This use of social network analysis is a simple exercise in graph theory and results of this experiment were as expected: social network metrics were able to determine which connections were most important in the computer network.

2.2.1.3 PieSpy: IRC and Shakespeare.

A clever program called PieSpy [22] monitors the messages in an internet relay chat (IRC) chatroom and gathers social networking data on the fly. This method of data gathering can be used to glean social networking data out of dialogues and was tuned to gather data from Shakespearean plays as shown in Figure 6.

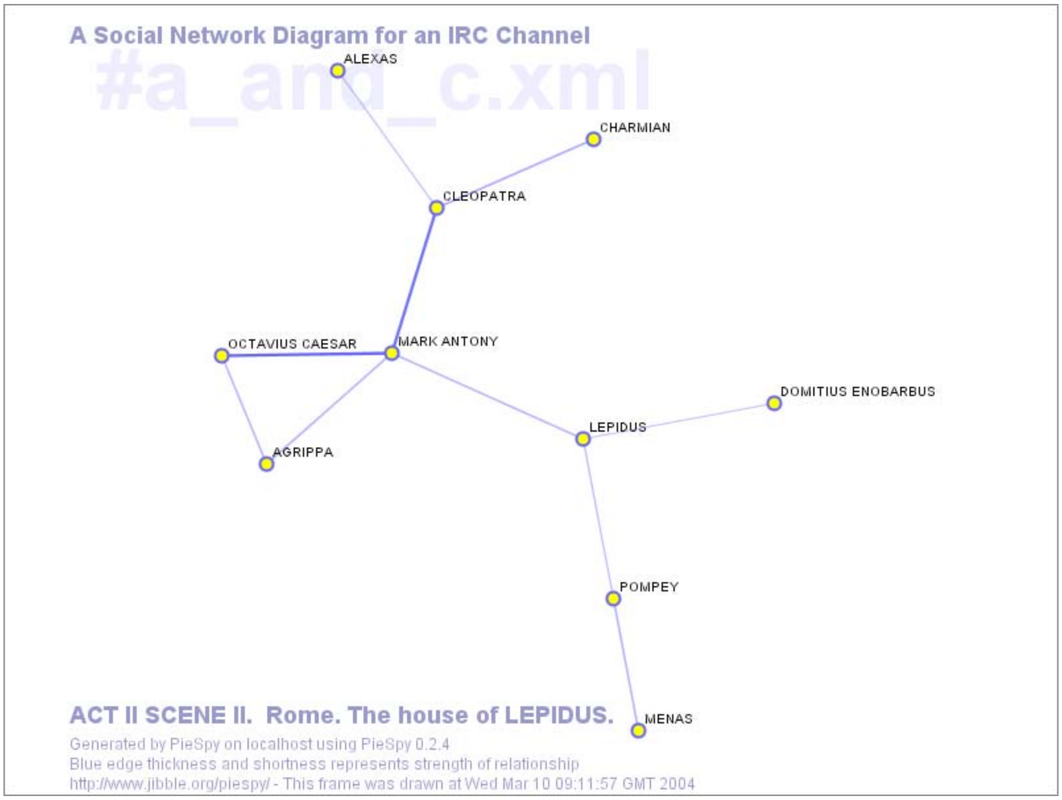

Figure 6: Social Network Data from Antony and Cleopatra Gathered by PieSpy [22]

PieSpy, while seemingly trivial, demonstrates the capability of creating social network data very quickly by mining a history of CMCs.

2.2.1.4 Traversing Social Networks and Mining Data from Email.

Hewlett-Packard Laboratories did research on finding the best methodologies in searching social networks [1]. In the research, "small world" or local search heuristics were compared to informed global search heuristics, and no social network analysis was performed on the data. It was found that in general, it is faster to pass a message by sending the message to an actor most similar in attributes to the intended recipient than to the adjacent actor with the highest degree.

What was most interesting about this research was how social network data from was created from email logs. Email logs are a readily available source of data when given the permission of the users. No further information is given about the

methods they used to gather the network data from the email logs, nor the format or any specifications of the email logs. Regardless, mining email logs is a viable method social network data collection and warrants further study.

2.2.1.5 Uncloaking Terrorist Networks.

Krebs also used social network analysis to "uncloak terrorist networks" after the terrorist attacks of 2001 [16]. By gathering data from news articles that followed the attacks, Krebs was able to construct a sociogram representing the terrorist network as shown in Figure 7.

After some investigation, it was alleged that Mohammed Atta was the leader of the covert operation. Krebs took the network centrality metrics of degree, closeness, and betweenness, and found that Atta had the highest score for all three metrics. These social network metrics support the idea that he is the leader of the operation.

Before jumping to conclusions, Krebs is careful to warn that social network analysis is not necessarily a detector of criminals. However, social network analysis can help determine structure and importance of members of a society.

2.2.1.6 Potential Insider Threat Mitigation.

Studying user behavior based on social network analysis metrics could potentially provide means of characterizing the insider threat. Changes in social network analysis metrics indicate a change in an actor's email messaging behavior. As Shaw writes, a change in behavior may be indicative of a potential insider threat [30].

Some social network analysis metrics that may be useful are Freeman Density, Bonacich Power, and k-cores. These metrics are standard in the social network analysis software programs UCINet and Pajek. Freeman Density is a measure of the centrality of an actor based on the number of actors connected to it. Bonacich Power is a metric taken to demonstrate the ability of social network analysis to provide information about actor power and centrality. A k-core is a loose group of actors in which

more tightly-knit groups of actors are found. Additionally, analyzing the changes to a specific actor's ego network also indicates changes to an actor's behavior. [11].

2.2.2 *Difficulty in Gathering Social Network Data.*

While the information that social network analysis can provide is helpful, gathering social network data is a difficult task. Gathering social network data is even more difficult when dealing with large populations and over protracted periods of time. Standard methods of data collection include conducting interviews/surveys, observing the actors, or extracting data from archived records.

The interview or survey-based approach to collecting data is extremely time consuming and not possible in all situations. Interviews and surveys inconvenience the people being studied (and can potentially invade their privacy), especially if they need to be repeated for a longitudinal study. Additionally, the questions asked are fairly simple and are only taken in one context. Further, these questions themselves often lead to bounding the number of connections, sometimes asking to name only a certain number [18]. In addition, these questions often do not capture the relative weight of the relationship and only the presence of a tie. Moreover, resources must be expended to carry out the interviews and surveys. The Network Roundtable at the University of Virginia developed a tool that can generate social network data from the results of customizable online surveys. While this expedites the process, it still requires user interaction and is only as accurate as the person filling out the survey [34].

Resources required to observe a social structure with people are especially high and time consuming. Observation often requires getting the permission of those studied, a permission that is not always granted. Moreover, the observers can only gather so much information, and this method of data collection works best when studying relatively small groups of people with close interaction [35].

On the other hand, the cost of gleaning social network data from archived information requires no intense observation of live subjects. Information can be gathered

from lots of different sources such as newspapers, attendance records, or email traffic [16]. Gathering data from recorded archives is done in a short period of time as opposed to gathering data as it happens, making it much easier to perform longitudinal studies. However, it does require time to read the archived data and get out the important information. Moreover, as archives are records of the past, they often do not provide information about the current social structure [35]. In terms of potentially mitigating the insider threat, collecting data archives is a viable option if the data can be collected and analyzed in a timely manner.

2.3 Summary

The insider threat is a significant risk that costs businesses millions of dollars annually. Current tools are inadequate and and research efforts toward mitigating its risk show that it is very difficult and potentially impossible to detect or characterize malicious insiders. The best method of mitigating the risk of the insider threat is through prevention and keeping pertinent information about employees.

The study of social networks has profoundly influenced the fields of mathematics, statistics, and economics as well as the fields of sociology, and psychology [35]. Social network analysis can provide useful information about a network and the actors within it. It is believed that the use of computers can make social network analysis feasible and potentially aid in mitigating the insider threat. While the impact of social network analysis on mitigating the insider threat remains to be seen, no studies can be done without social network data.

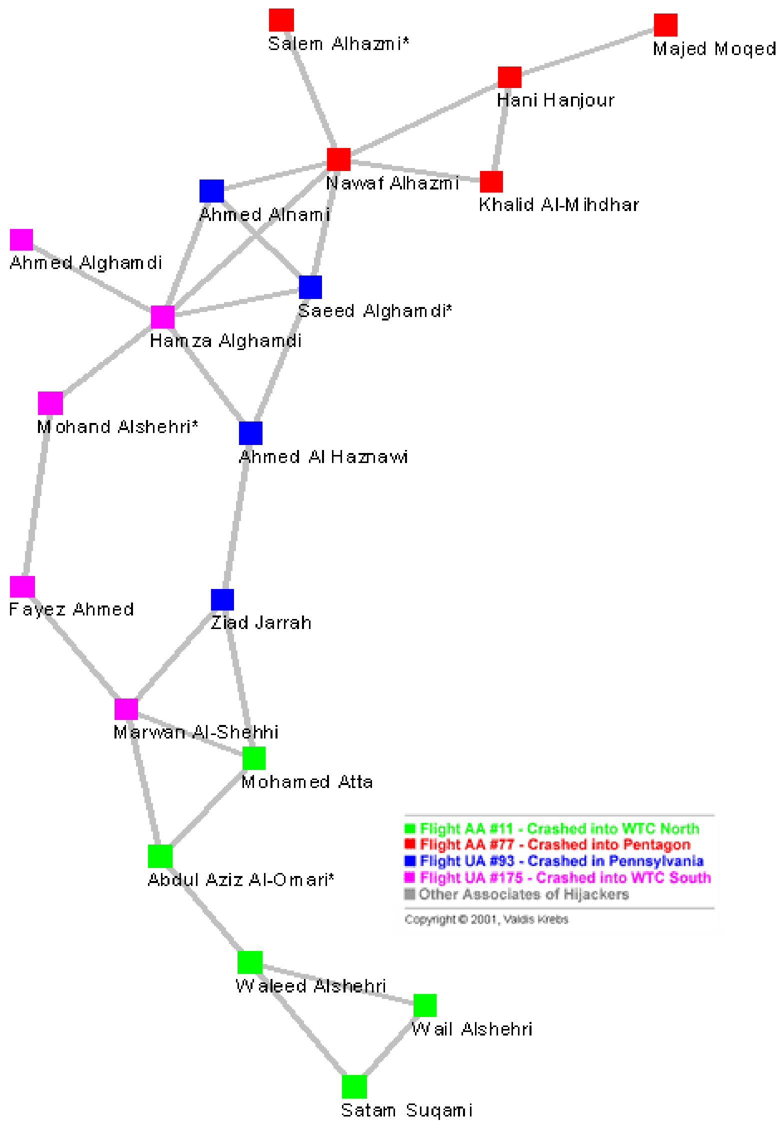

Figure 7: Krebs' Mapping of the 2001 Terrorist Network [16]

III. System Implementation

The objective of this research is to show that the creation of useful social network data from Simple Mail Transfer Protocol (SMTP) data in a time-efficient manner is possible.

3.1 Overview

It is possible to automatically create social network data that can be read and analyzed by current social network analysis tools. The metrics gathered by these tools may prove useful to social network analysts in characterizing organizational behavior. These characterizations can be used in a tool to provide supervisors more information about their subordinates. Since human behavior is reflected in email usage, future research may show that good social network data can be mined from SMTP logs.

Creating social network data from readily available SMTP logs is cheaper than taking surveysconducted on company time. The automatic creation of social network data also allows social network analysts to study the short-term dynamics of a large set of actors, impossible to do with current social network data gathering methods. Unfortunately, it is unknown how quickly a system can create social network data with these methods.

This chapter describes the proof of concept system developed to achieve research objectives and covers the development of the system, the system components, and the operation of the system.

3.1.1 System Overview.

To create social network data from SMTP logs, data is gathered, filtered and/or parsed, and mined for information. If privacy is required, the data can also be anonymized. The process enumerated below is illustrated in Figure 8.

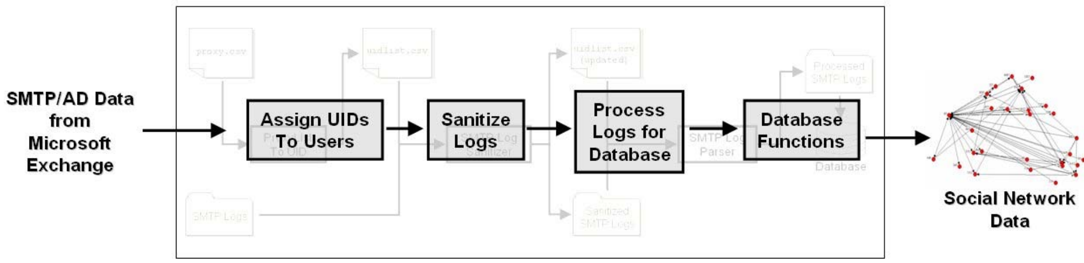

Figure 8: System Process and Components

1. Gather SMTP/Active Directory data with MS Exchange
2. Enumerate Users with a Proxy List
3. Sanitize Data by Assigning Unique Identifiers
4. Process Sanitized Data for a Database Import
5. Import Processed Data and Generate Social Network Data

Pre-existing tools are used to perform the first step as SMTP logs and user information are generated automatically by Microsoft Exchange and are extracted with Javelina's Advantage. To be useful, the output from the system is readable by current social network analysis software like UCINet or Pajek, as it is far too time-consuming to perform social network analysis on a network of over 1000 actors by hand.

Thus, to achieve the goals of this research, a system that performs the missing steps of the process is implemented. As specified above, the generation of useful social network data from SMTP logs requires components that provide the following four services:

1. Assign UIDs to Users with a Proxy List
2. Sanitize Logs by Replacing Usernames with UIDs
3. Process Sanitized Logs for Database Import
4. Database Functions to Import Data and Generate Social Network Data

Each of these functions are encapsulated in a component implemented for the purposes of this research as shown in Figure 9. The `ProxyListToUID`, `SMTPLogSanitizer`, and `SMTPLogParser`programs are written in Java 1.50 and require Java JRE version 1.5. Queries are made with MySQL version 4.1. Microsoft Exchange and Javelina ADVantage generate the proxy list file `proxy.csv` and the SMTP logs. The implemented `ProxyListToUID` component assigns a unique identification number (UID) to each user as denoted by a proxy list. The implemented `SMTPLogSanitizer` component sanitizes the logs by replacing user names with UIDs. The implemented `SMTPLogParser` processes the sanitized logs and creates database importable information. Finally, database functions import the data from the `SMTPLogParser` and generate social network data that are readable by current social network analysis programs.

3.1.2 Expected Usage.

The system is expected to be given a fresh set of SMTP logs every week or month to sanitize, process, and add to the database. This relatively short time interval ensures that the social network data recorded is current and thus presents a more accurate description of the behavior of the system users.

3.1.3 Privacy Protection.

To protect the privacy of users, access to certain sensitive files must be controlled. This system was created with the assumption that the organization whose

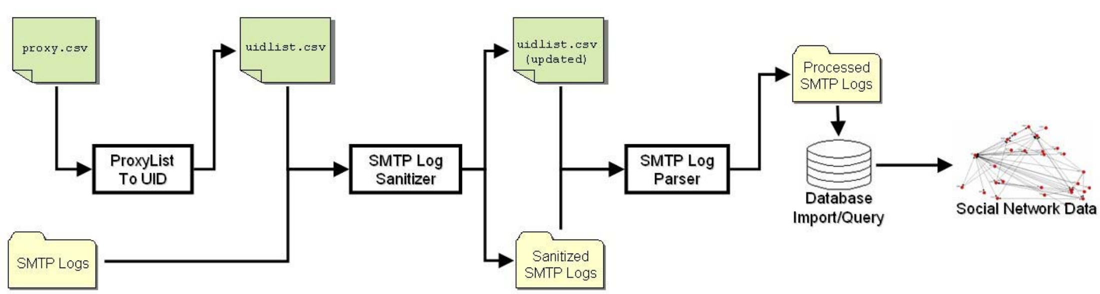

Figure 9: System Implementation

email traffic generated the original SMTP log data is planning to give either the sanitized data or the parsed-sanitized data to an external party. This external party is expected to analyze the generated social network data. The organization owning the original SMTP data can possess all of the data objects created.

This access scheme to protect the privacy of users is shown in Figure 10, where sensitive files that could violate the privacy of users are in shaded boxes. These boxed files are not to be accessed by external parties.

In summary, access to data objects should be given as follows:

- Private Data, exclusive to Owner of SMTP Data
  - SMTP Logs
  - proxylist.csv
  - uidlist.csv
- Sanitized Public Data, External Entities can Possess
  - Sanitized SMTP Logs
  - Parsed-sanitized Logs
  - Databases/Social Network Data

3.2 Gather SMTP and ProxyList Data

Correctly formatted SMTP log data is mined for information. Setting up a Microsoft Exchange server to gather SMTP data for this experiment is simple. The

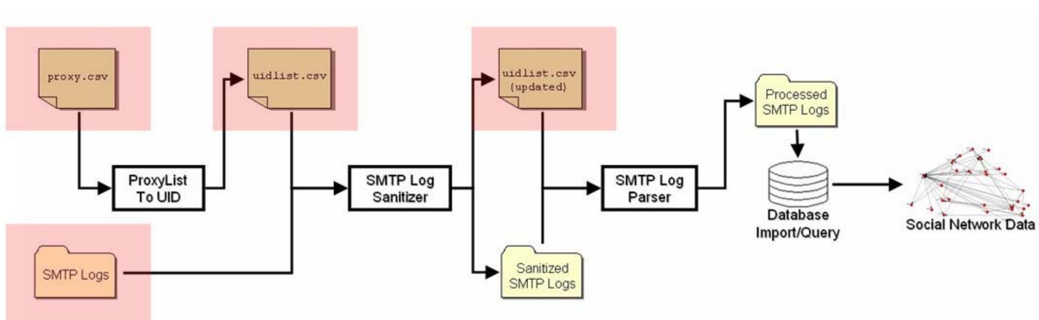

Figure 10: File Permission to Protect Privacy

SMTP log files are in the NCSA Common Log File format and become the input for the sanitization component.

3.2.1 *Input.*

Outlook Exchange Server, Exchange System Manager

3.2.2 *Output.*

Directory of `*.log` files.

3.2.3 *Proxy List.*

The proxy list file `proxy.csv` is generated using the Javelina ADVantage tool. This tool goes through the Active Directory and gathers the aliases, emails addresses, and X400 information about every user and creates output as shown in Figure 11.

3.3 *ProxyListToUID Component*

The `ProxyListToUID` component resolves the multiple aliases of an actor to a specific actor. For example, a user in the system named *Jason Yee* might use both the email addresses `jyee@afit.edu` and `jason.yee@afit.edu` . This component assigns the same identity to the two different email addresses. While this is not possible for outsiders to the system, it is definitely possible to identify users in an

```
Yee Jason 2dLt AFIT/ENG,Jason.Yee@afit.edu,
"X400:c=US;a= ;p=AFIT;o=HANGAR;s=Yee;g=Jason;
smtp:jyee@afit.edu
SMTP:Jason.Yee@afit.edu"
"Smith John A Civ AFIT/SC",John.Smith@afit.edu,
"X400:c=US;a= ;p=AFIT;o=HANGAR;s=Smith;g=John;
smtp:jasmith@afit.edu
smtp:jsmith1@afit.edu
SMTP:John.Smith@afit.edu
SMTP:John.Smith.1@afit.edu"
...
```

Figure 11: Proxy List File Format

email system. This information can be gathered from the Active Directory using a tool called ADVantage from Javelina Software.

This user information can be used to assign a unique identification number (UID) to each *individual user* in the system. It is important to understand that a UID is meant to correspond to a single human user. It is essential to identify unique users if social network analysis results are expected to be accurate. This is something that is taken for granted when carrying out interviews and surveys to gather social network data.

The `ProxyListToUID` program assigns a unique user identification number (UID) to each listed user. Extra email aliases are mapped to the same UID.

3.3.1 Usage.

The one argument required to call the `ProxyListToUID`is the full path to the proxy.csv file. From the folder containing the class files, enter:

```
java ProxyListToUID ‘‘c:\full-path-to\proxy.csv’’
```

3.3.2 Input.

The `ProxyListToUID` program requires a proxy list file `proxy.csv` as input. The proxy list file should contain entries of all of the users with the form:

```
DisplayName,EmailAddress,ProxyAddresses
```

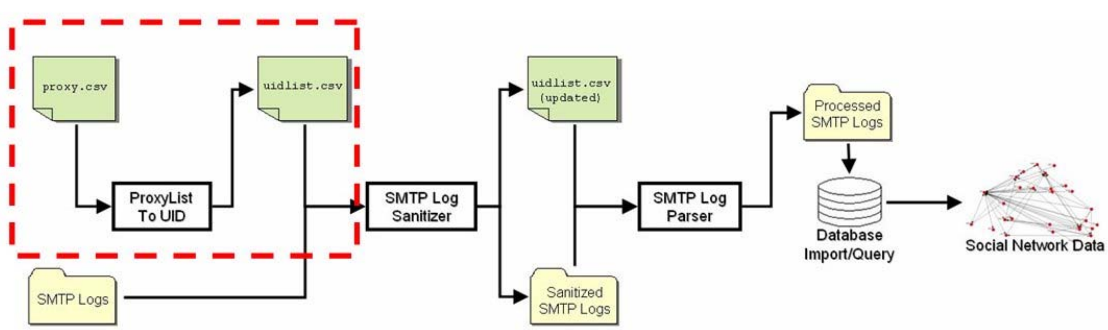

Figure 12: ProxyListToUID Component

An example of an entry is shown in Figure 11. This is the same format used in exporting Active Directory contents by Javelina's ADVantage tool. The string for the proxy file follows Microsoft convention and may therefore have spaces. Thus, the string should include quotation marks.

3.3.3 Output.

The `ProxyListToUID` program produces the UID list file `uidlist.csv` as a list of user IDs assigned to email addresses. The `uidlist.csv` file in the form:

```
UniqueID,EmailAddress
```

The correct output for the sample input in Figure 11 is shown in Figure 13. It is important to note that the output file `uidlist.csv` is considered private information that can be used to un-sanitize the data.

```
0,Jason.Yee@afit.edu
0,jyee@afit.edu
1,John.Smith@afit.edu
1,jasmith@afit.edu
1,jsmith1@afit.edu
1,John.Smith.1@afit.edu
...
```

Figure 13: Format of UID list

When the program is run in a terminal window, it will display the name of the file, number of email addresses imported, and the time it took as shown in Figure 14.

```
Converting "proxy.csv" to UID list
uidlist.csv
uidlist.csv created
282ms to complete.
```

Figure 14: Terminal Output for ProxyListToUID

3.4 SMTPLogSanitizer Component

For the purposes of this research, specific identities were not required nor desired. Thus, before the SMTP logs are analyzed, they are sanitized. The sanitization process used in this research makes SMTP logs relatively safe for distribution by replacing the user name of an email address with a number, thus masking the identity of the user. While it is possible to also hide the domain name, that information was retained so that other researchers are able to determine which users are internal and external to the system. It is important to note that NCSA formatted SMTP logs do not contain any information about the subject or content of the email message. Thus, an SMTP log is considered sanitized if the user names are anonymized.

The `SMTPLogSanitizer` program keeps a record of the email addresses already seen and the number of unique users identified. If the log sanitizer finds an email address that it has already seen, it replaces that email address with the UID associated with it. If the log sanitizer finds an email address that it has not seen, it generates a new UID, assigns it to the email address, and replaces the email address with the UID. The log sanitizer program `SMTPLogSanitizer` was built with those requirements.

To ensure correct performance, it is extremely important that a record of which UID corresponds to which email address is kept. Otherwise, running the `SMTPLogSanitizer` on another log file may assign different email addresses the same UID, defeating the purpose of UIDs.

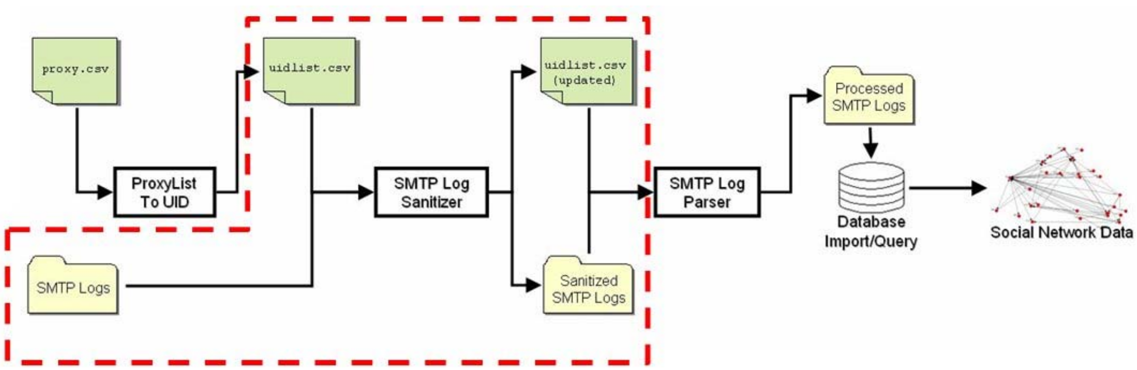

Figure 15: SMTPLogSanitizer Component

3.4.1 Usage.

If a the ProxyListToUID output file uidlist.csv is available, move it into the folder containing SMTPLogSanitizer class files. SMTPLogSanitizer looks for a file named uidlist.csv in its working directory; if no file is found, a new user ID list is created. Enter this command to run the program:

```
java SMTPLogSanitizer ‘‘c:\full-path-to\SMTPLogs\’’
```

3.4.2 Input.

SMTPLogSanitizer takes a folder containing *.log files as explicit input. The file uidlist.csv is implicitly input. This component will find all *.log files in the folder and create sanitized versions of them, as well as update the UID list file uidlist.csv with the new addresses in the log files.

3.4.3 Output.

The SMTPLogSanitizer program sanitizes each *.log file by replacing the user portion of all email addresses with a UID. For instance, if the email address joe.user@afit.edu is not in the UID list, it will be replaced with 27182@afit.edu in the sanitized log file if 27182 is the next UID generated. The new association between the email address and a UID is added to the UID list uidlist.csv located in the working directory.

If the email addresses jane.user@afit.edu and juser@afit.edu are both assigned the UID 3141 by the ProxyListToUID program, either occurrence is replaced with 3141@afit.edu in the sanitized log.

A new file with all of the substitutions is created with the filename s_ concatenated in front of the original file name. This file is placed in the SanitizedLogs subfolder of the working directory, and the original *.log file is unchanged.

The SMTPLogSanitizer program outputs information to the terminal about the speed and size of read files. File name, lines in the file, time taken to sanitize the file, speed, and number of UIDs are given as shown in Figure 16.

```
01Nov.log
Reading 01Nov.log
.   1531256 lines read in 01Nov.log
.   177031ms, 8.649648931543062 lines/ms
.   23363 new UIDs added, 24913 total
UIDs
```

Figure 16: Terminal Output for SMTPLogSanitizer

For example, the sanitization of file `2004-10.log` results in a file named `s_2004-10.log` in the `SanitizedLogs` subfolder of the folder the `SMTPLogSanitizer` class files are in, as well as an update of the UID list `uidlist.csv`.

Now that sanitized copies of the logs have been created, they are used without fear of invasion of privacy. The `SanitizedLogs` folder can be distributed, while the original SMTP logs and the UID list `uidlist.csv` are to remain secured by the system administrators.

3.4.4 Implementation Notes.

The `SMTPLogSanitizer` program requires a persistent store for the the email addresses and the UIDs that they correspond to. The current form stores this data in plaintext in the file `uidlist.csv`. When this program is called, the UID list is parsed and imported into a hashtable, and before the program exits, the contents of the hashtable are rewritten to the UID list. As a result, `SMTPLogSanitizer` is reusable.

An issue with the `SMTPLogSanitizer` is the presence of invalid email addresses, which are not sanitized. Valid email addresses have a username, the '`@`' character, and top level domain or country code like `.com`, `.net`, or `co.uk`. Email addresses must consist of the alphanumeric characters and the characters '`.`', '`_`', '`%`', and '`-`'. Invalid email addresses contain incorrect characters in the user name such as '`"`', '`'`', '`/`', '`+`', or '`=`'; these are most likely the result of poorly written scripts.

These invalid email addresses are ignored in this implementation, but the code can easily be changed to have them discarded. Since invalid email addresses are not

accepted by the next component and incorporated in the social network data, they are not considered to be an issue.

3.5 SMTPLogParser Component

The `SMTPLogParser` component extracts data from SMTP logs that are imported into a database. Useful information from SMTP logs are the date and time an email was sent, the sender and recipient, if the sender and recipient are internal, and how many recipients received the same email. This information is used by the database to generate social network data. The `SMTPLogParser` program was developed to extract such data from sanitized NCSA formatted SMTP logs.

To respect the privacy of the email users, this extraction process is done on sanitized logs where the sender and recipient are numbers and the domain name is used only to determine the internal or external status of the user

3.5.1 Usage.

The one argument required to call the `SMTPLogParser` program is the full path to a folder containing sanitized logs. From the folder containing the class files, enter:

```
java SMTPLogParser‘‘c:\full-path-to\SanitzedLogs\’’
```

3.5.2 Input.

The files in the input folder *must be* sanitized logs. This means that all email ad-

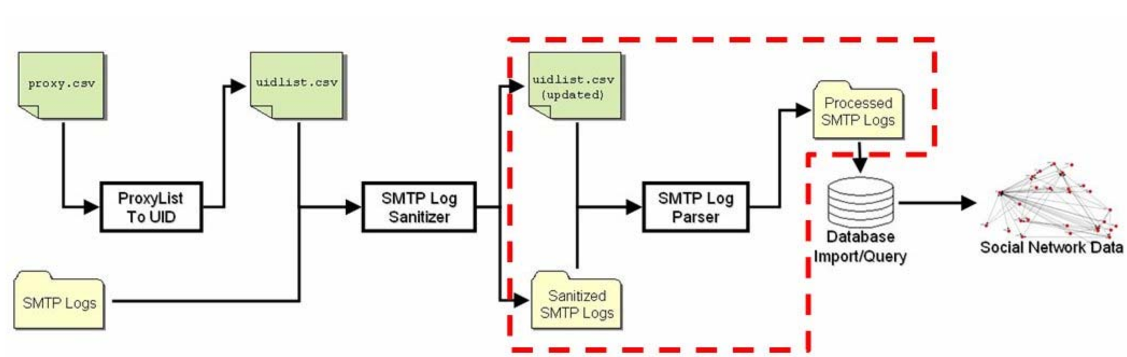

Figure 17: SMTPLogParser Component

dresses in the files must be in the form of `<UID>@domain`. The path to a `SanitizedLogs` folder created by the `SMTPLogSanitizer` program will work perfectly.

3.5.3 Output.

The `SMTPLogParser` program parses the contents of each sanitized log file and and extracts the date, time, sender UID, recipient UID, whether the sender and recipient are internal, and the number of recipients in the email. For example, the input of Figure 18 results in the output shown in Figure 19

```
129.92.1.65 - OutboundConnectionCommand [09/Dec/2004:08:01:17 -0500]
"MAIL -?FROM:<2718@afit.edu> SMTP" 0 4
129.92.1.65 - OutboundConnectionResponse [09/Dec/2004:08:01:17 -0500]
"- -?250 2.1.0 2718@afit.edu....Sender OK SMTP" 0 43
129.92.1.65 - OutboundConnectionCommand [09/Dec/2004:08:01:17 -0500]
"RCPT -?TO:<1828@afit.edu> SMTP" 0 4
129.92.1.65 - OutboundConnectionCommand [09/Dec/2004:08:01:17 -0500]
"RCPT -?TO:<4590@ieee.org> SMTP" 0 4
```

Figure 18: Sample Sanitized Log Input for SMTPLogParser

```
2004/12/09 08:01:17 2718 1828 1
1 2
2004/12/09 08:01:17 2718 4590 1
0 2
```

Figure 19: SMTPLogParser Output for the Sample in Figure 18

The columns in the output correspond to the date, time, sender UID, recipient UID, internal status of sender (`1` if internal, `0` if external), internal status of recipient, and the number of recipients. The internal status will be `1` if the domain is as specified (`afit.edu` in this case) and `0` otherwise.

Terminal output shows the time taken, speed of parsing, and the number of messages read per file and the total time taken to parse all files as shown in Figure 20.

```
s_01Nov.log Reading s_01Nov.log
70797ms, 13.091190304673926
lines/ms .   195930 messages
recorded ...  1397954ms to
complete.
```

Figure 20: Terminal Output for SMTPLogParser

3.5.4 Implementation Notes.

The original component did not include a way to count the number of recipients per message. The current version counts the number of *valid* recipients. The number of recipients is counted so that "spam" or "mass-emails" are ignored.

As written, `AFIT` is hard-coded in to determine what email addresses are internal. This can easily be changed in the source code by changing the `INTERNAL_DOMAIN` variable.

The `SMTPLogParser` depends on the `SMTPLogSanitizer` to assign the same UID to the different addresses of each user. `SMTPLogParser` will parse whatever data is given: it is recommended that the `SMTPLogSanitizer` program is used to sanitize the data.

Multiple instances of the `SMTPLogParser` component can be run in parallel to process data. This can greatly expedite the processing of sanitized SMTP log data.

3.6 Database Functions Component

The processed information provided the needed information to create social network data. All that remains is to setup the database, import the data from the processed logs, and export the query results.

3.6.1 Database Setup.

Before importing the data, a table carrying the values is created. The fields in the table are the same as the fields created by the `SMTPLogParser` program including

a primary key:

- MessageID (database key)
- Date
- Time
- UID of Sender
- UID of Recipient
- Internal status of Sender
- Internal status of Recipient
- Number of Recipients of Message

The query used to create the message table as described above is shown in Figure 22. Creating the table needs to be done only once.

3.6.2 Importing Data into Message Table.

When the database and required table is prepared, the data extracted by the `SMTPLogParser` is imported. The query used to import data is shown in Figure 23.

3.6.3 Creating Sociograms.

Finally, social network data that is analyzed by UCINet is extracted from the `SMTPLogParser` generated data in the database. This is a two-step process that requires a query and the addition of a header.

3.6.3.1 SQL Query Results.

In the first step, a MySQL query is made and the results are exported. By

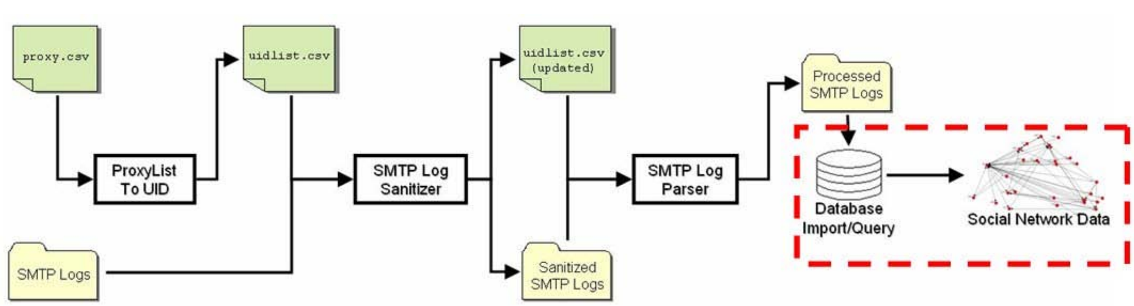

Figure 21: Database FunctionsComponent

```
### Create EmailMessageLog create table
‘emailmessagelog‘.‘EmailMessageLog‘ (
‘MessageID‘ int NOT NULL AUTO_INCREMENT
, ‘Date‘ date NOT NULL , ‘Time‘ time
NOT NULL , ‘SenderUID‘ mediumint NOT
NULL , ‘RecipientUID‘ mediumint NOT
NULL , ‘InternalSender‘ bool NOT NULL
, ‘InternalRecipient‘ bool NOT NULL ,
‘NumRecipients‘ mediumint NOT NULL , PRIMARY
KEY ( ‘MessageID‘ ) )
```

Figure 22: SQL Query to Create Email Message Table

```
### Import Parsed-Sanitized Log
load data local infile ’ps*.txt’
into table emailmessagelog
fields escaped by ’\\’
terminated by ’\t’
lines terminated by ’\n’
(
‘Date‘,
‘Time‘,
‘SenderUID‘,
‘RecipientUID‘,
‘InternalSender‘,
‘InternalRecipient‘,
‘NumRecipients‘
)
```

Figure 23: SQL Query to Import Data from SMTPLogParser

altering the query, different social network maps are generated. In this research, the sociograms are exported with space-delimited fields. This is done easily with a program like SQLYOG[1]. A sample of a sociogram extracted in UCINet edgelist format without header information is shown in Figure 24.

There are many different ways to calculate tie strength, but for the purposes of this experiment, tie strength is calculated as the number of email messages sent from the sender to the recipient. For example, in Figure 24, actor `0` sent five emails

[1]SQLYOG exports in UNIX-formatted text and UCINet only accepts text in PC format

```
0 1 5
0 2 14
0 3 16
1 0 8
1 3 13
2 1 11
2 3 6
3 2 12
...
<SenderUID> <RecipientUID> <TieStrength>
```

Figure 24: Exported Query Results

to actor `1` while actor `1` sent eight emails to actor `0`. This is directed, non-symmetric social network data.

Altering the query allows the user to create sociograms based on select data extracted from the SMTP logs. Figures 28-31 show different MySQL queries and describe the sociograms created.

3.6.3.2 Header Information.

The second step involves inserting the following header information at the beginning of the file. The exported file is edited and the header shown in Figure 25 is inserted before the data. UCINet will automatically calculate the number of distinct actors so a specified number of actors (currently `n = 1`) can be input as shown.

```
dl
n = 1
format = edgelist
labels embedded
data:
```

Figure 25: Header for Top of Exported Query Results

Once this header is given, this file contains social network data in a mostly valid DL format file. Upon importing this file into UCINet, an error message will pop up as shown in Figure 26. This error message will determine what number of nodes, the number that should replace the `1` currently in the header. In the case of Figure 26, `n = 1491`. Given this information, the DL format file can be corrected as shown in

Figure 27. Now, UCINet can import and perform social network analysis upon this valid social network data.

Figure 26: UCINet Import Error

```
dl
n = 1491
format = edgelist
labels embedded
data:
```

Figure 27: Header for Top of Exported Query Results with 1491 Nodes

3.7 Summary

This chapter described the proof of concept system developed to achieve research objectives. The next chapter discusses the methodology used to test this system.

```
SELECT SenderUID, RecipientUID, COUNT(*)
FROM emailmessagelog
GROUP BY SenderUID, RecipientUID
```

Figure 28: SQL Query to Create a Sociogram

```
SELECT SenderUID, RecipientUID, COUNT(*)
FROM emailmessagelog.emailmessagelog
WHERE InternalSender = 1 AND InternalRecipient = 1
GROUP BY SenderUID, RecipientUID
```

Figure 29: SQL Query to Create an Internal Sociogram

```
SELECT SenderUID, RecipientUID, COUNT(*)
FROM emailmessagelog.emailmessagelog
WHERE
InternalSender = 1
AND InternalRecipient = 1
AND Date BETWEEN "2004-09-01" AND "2004-09-31"
GROUP BY SenderUID, RecipientUID
```

Figure 30: SQL Query to Create an Internal Sociogram Limited by Date (September 2004)

```
SELECT SenderUID, RecipientUID, COUNT(*)
FROM emailmessagelog.emailmessagelog
WHERE
InternalSender = 1
AND InternalRecipient = 1
AND NumRecipients < 20
AND Date BETWEEN "2004-09-01" AND "2004-09-31"
GROUP BY SenderUID, RecipientUID
```

Figure 31: SQL Query to Create an Internal Sociogram Limited by Date (September 2004) and Number of Recipients (less than 20)

IV. Experiment Methodology and Results

This chapter documents the experiment methodology and the test results gathered. The second section discusses the results and findings of the experiment.

4.1 Experiment Methodology

The first section of this chapter describes the methodology of the experiments used to test the system implemented as specified in the previous chapter.

4.1.1 Goals and Hypothesis.

The intent of this research is to determine if the creation of useful social network data from Simple Mail Transfer Protocol (SMTP) data in a time-efficient manner is possible. This experiment determines the time required to create useful social network data out of readily-available SMTP logs and the usefulness of the generated data from the system described in the previous chapter.

It is expected that social network data is generated in under a day, the approximate time it takes to complete a computer-based survey. The social network data generated is expected to be in a form usable by UCINet and is imported and analyzed as described in the previous chapter. However, it is unknown exactly how long it takes to generate the social network data.

4.1.2 Evaulation Metrics.

The metrics used to evaluate the social network data generating system in this experiment are usefulness and timeliness.

The usefulness metric is the ability for the generated data to be read by the standard social network analysis program UCINet. UCINet is the most commonly used social network analysis program and files that can be read by UCINet can be read by almost all other social network analysis programs like Pajek and NetDraw. Thus, for this performance metric, if UCINet can take some standard social network analysis metrics from the generated social network data, the system is useful.

The metric of usability has no connection with the quality of data. This is primarily because no study on this kind of data has been done at the time of this research. Additionally, it is not within the scope of this research to analyze the social network metrics of the data.

The timeliness metric is the duration of time taken by each component to provide its service and the overall time required to generate social network data. If the total time to generate social network data is less than a day, the system's performance is considered to be good. Again, this is in comparison to the time required to gather social network data from a computer-based survey.

For future application in mitigating the insider threat, the timeliness metric is dependent on the "window of opportunity" within which a manager is able to prevent an insider incident through intervention. To get useful and timely information to a manager, the system must generate useful social network data (as determined by social network analysts) within that time period.

4.1.3 System Boundaries.

The system under test is the system that generates useful social network data through the process illustrated in Figure 32.

The scope of this experiment is limited to the process of creating useful social network data in a timely manner. It does not include the systems that generate the

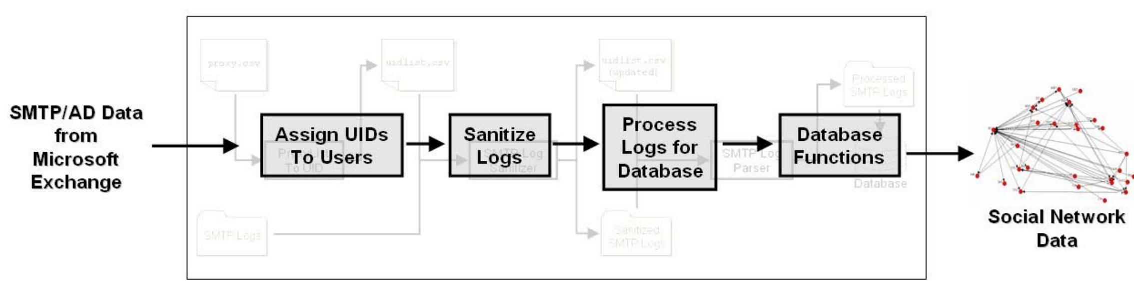

Figure 32: System Under Test

original proxy lists and SMTP data (Microsoft Exchange/Javelina ADVantage) or analyze the social network data (UCINet, NetDraw).

4.1.4 System Setup.

This testing is performed on a dedicated Dell Poweredge Pentium 4 hyperthreading-enabled 3.2 GHz computer with 2 GB RAM and the Windows XP Professional Edition operating system. The system tested is the same system implemented in the previous chapter, with components developed in Java 1.5, and a MySQL 4.0.21 server.

4.1.5 Workload.

The workload in this research consists of a list of users and their different email addresses and SMTP logs in the National Center for Supercomputing Applications (NCSA) format [19]. To obtain realistic data for a medium-sized organization (1550 members), data was collected from AFIT's Microsoft Exchange servers over 86 days during the months of October, November, and December of 2004. The SMTP data consists of approximately 3.6 gigabytes worth of text as shown in Figure 4.

October is the first month in the Fall quarter and relationships between students and faculty are still forming. However, this quarter follows the primer courses given in September so the students have had over a month to meet. Thus, in October, relationships are somewhat formed and more stable. November and December are the final two months in the Fall quarter and should exhibit generally similar behaviors. The data provided is missing the last week in October because of a server crash.

Month	Days	Size
October	25	1,034,273 KB
November	30	1,385,098 KB
December	31	1,359,860 KB

Table 4: Data Size

4.1.5.1 ProxyListToUID Workload.

The `ProxyListToUID` component is tested with the proxy list created from

AFIT's Active Directory. This proxy list is 259 KB and contains the information for 1550 distinct users, each with multiple email aliases.

4.1.5.2 SMTPLogSanitizer Workload.

The `SMTPLogSanitizer` component is given the workload of three months of SMTP data in the form of 12 separate log files in a directory. The UID list, the result of the `ProxyListToUID` component is also part of the workload for this component.

4.1.5.3 SMTPLogParser Workload.

The `SMTPLogParser` component is given the workload of three months of sanitized SMTP data in the form of 12 separate sanitized log files in a directory. These sanitized logs are the output of the `ProxyListToUID` component.

4.1.5.4 Database Queries.

The database functions component is given the workload of importing three months of processed SMTP log files, the output of the `SMTPLogParser` component.

The database then has a workload of 12 queries to create different sets of social network data. These queries will vary by date range and restriction. The possible date ranges and restrictions are as follows:

- Date Range
  - October
  - November
  - December
  - All months
- Restriction
  - None
  - Internal Only
  - Internal Only and Number of Recipients less than 20

4.1.6 Evaluation Technique.

First, the software is verified by testing and debugging, then validated on a small data set. The small data set is small and the results of the test are validated by hand. Then, the timeliness and usefulness of the system is evaluated by creating social network data from a workload consisting of real data. The system is tested in a the manner it is expected to operate, sanitizing and processing one month of SMTP log data at a time. Times are gathered by direct measurement.

4.1.6.1 Verification of Components.

Verification that the components are performing correctly is tested by debugging and white-box code walkthroughs.

Validation of the results is done by testing system with artificially-generated SMTP data. The social network data of the artificial SMTP data is gathered manually and is compared to the social network data automatically generated by the system. The components are correct if they provide the expected output. This method of evaluation is justified as no similar system with similar data has been implemented.

This portion of testing is done during the development of the system and will not be reported in the results and findings.

4.1.6.2 Evaluating Timeliness.

The time used to generate social network data consists of direct measurement of the durations of multiple runs of each component. The results cannot be validated analytically or by simulation or as no system like this has been created.

To measure the runtimes, the full process of creating social network data out of SMTP logs is repeated three times and the time spent on each component is recorded.

4.1.6.3 Usability of Social Network Data.

Validation of the usability of the social network data is evaluated by importing it into UCINet and taking standard social network analysis metrics. The metrics

taken are the Freeman Density, Bonacich Power, and k-Cores social network metrics. Ego networks are also extracted.

These three metrics were chosen because they provide useful information on monitoring the behavior individual users. Studying email messagaging behavior could potentially provide means of characterizing the insider threat. Changes in these metrics indicate a change in an actor's email messaging behavior. As Shaw writes, a change in behavior may be indicative of a potential insider threat [30]. Freeman Density is a measure of the centrality of an actor based on the number of actors connected to it. Bonacich Power is a metric taken to demonstrate the ability of social network analysis to provide information about actor power and centrality. UCINet finds k-cores, which are loose groups of actors in which more tightly-knit groups of actors are found [11]. Changes to ego networks show that the actors that a certain actor interacts with are changing.

4.1.7 Summary of Experiment Methodology.

The experiments outlined in this chapter determine whether or not a system that automatically creates useful social network data out of SMTP data in a time-efficient manner is possible.

4.2 Results and Findings

The first subsection shows the time taken for each component to complete its task and provides commentary. In the next section, some social network metrics are calculated with UCINET with the output of the system.

4.2.1 System Timing Test Results.

The average time required for the components to process three months of SMTP data and generate social network data is about 80 minutes as shown in Table 5. On average, it takes about half an hour to add a month to the database and generate a set of social network data incorporating the new information as shown in Table 6.

Component	Runtime
ProxyListToUID	< 1 min
SMTPLogSanitizer	54 min
SMTPLogParser	24 min
Database Import	< 1 min
SQL Query	< 1 min
Total (three months)	≈ 80 min

Table 5: Component Runtimes for Three Months

It is apparent that this process takes much less than a day and the system can produce timely social network data. The majority of the time creating the social network data is spent sanitizing and processing the SMTP logs. The SQL queries all took seconds to run. The SQL queries used to import the parsed data took the most time.

Component	Runtime
ProxyListToUID	< 1 min
SMTPLogSanitizer	$\approx$20 min
SMTPLogParser	$\approx$10 min
Database Import	< 1 min
SQL Query	< 1 min
Total (one month)	≈ 30 min

Table 6: Component Runtimes for One Month

4.2.1.1 ProxyListToUID.

The `ProxyListToUID` program was given the proxy list as input. From this data, the program assigned 3454 email addresses to 1550 distinct users and wrote the output to the UIDlist in under a second as shown in Table 7. This component had little impact on the overall time to generate social network data.

4.2.1.2 SMTPLogSanitizer.

Over 170,000 total UIDs were assigned by `SMTPLogSanitizer` with an average

	Test 1	Test 2	Test 3
Runtime	282	281	282

Table 7: ProxyListToUID Runtimes in Milliseconds

time of about 55 minutes as shown in Table 8. This was the most time consuming of the components. The time spent sanitizing the SMTP logs from October was less than the other months because a server crash in October caused fewer days to be logged. There were 25 days logged in October, 30 in November, and 31 in December as shown in Table 4.

	Test 1	Test 2	Test 3
October	15.4	14.5	17.7
November	21.3	20.1	20.1
December	20.6	19.7	18.7
Total	57.3	54.3	53.5

Table 8: SMTPLogSanitizer Runtimes in Minutes

The time required to sanitize a month's logs appears independent. The time required to sanitize logs was expected to decrease as more email addresses were added to the UID list, but this did not happen. Many new email addresses (always senders) are created and added to the UID list every month as shown in Table 9. Many of the new UIDs were assigned to addresses like `BOUNCE-1234-1234124@fedweek.sparklist.com`, usually generated by mailing lists that users subscribe to.

	UIDs Added
October	57,232
November	49,590
December	67,902
Total	174,724

Table 9: UIDs Added by the SMTPLogSanitizer

4.2.1.3 SMTPLogParser.

`SMTPLogParser` processed three months of data in the average time of about

24 minutes as shown in Table 10. As is expected, the time spent parsing a month is independent. Each month, with about a million ties, took under 10 minutes to parse.

To clarify the difference between an email and a tie, when *Alice* sends an email to *Bob* and *Carol*, one email was created by *Alice* and two ties, $Alice \rightarrow Bob$ and $Alice \rightarrow Carol$, are created. Ties are used to generate social network metrics.

	Test 1	Test 2	Test 3
October	6.9	6.9	6.2
November	9.1	8.7	8.3
December	9.3	8.0	7.9
Total	25.3	23.6	22.4

Table 10: SMTPLogParser Runtimes in Minutes

This component was given the output of the `SMTPLogSanitizer` as input as specified in the Workload subsection. The `SMTPLogParser` mined the sanitized data for over 1.8 million individual emails for a total of over 3 million connections between users as shown in Figure 11. These numbers are consistent with the missing emails from a week of server downtime in October.

	Emails	Ties
October	464,895	883,024
November	675,411	1,149,305
December	689,071	1,065,800
Total	1,829,377	3,098,129

Table 11: Number of Emails and Ties

4.2.1.4 Database Creation and Import.

Creating the database table required to hold the data took under a tenth of a second. Importing one month of database-readable data generated by the `SMTPLogParser` program took about 10 seconds as shown in Table 12. This component had little impact on the overall time to generate social network data.

Action	Test 1	Test 2	Test 3
Create Table	0.078	0.047	0.062
Import October	7.765	7.719	7.750
Import November	10.95	10.922	10.812
Import December	10.282	10.265	10.313

Table 12: Database Creation and Import Times in Seconds

4.2.1.5 Generate Social Network Data from Database Queries.

As specified in the workload subsection of the previous section, 12 different sets of social network data were created. The query times required to generate all 12 different sets of social network data in the edgelist format took each less than six seconds as shown in Table 13. This component had little impact on the overall time to generate social network data.

Social Network Data	Execution Time
All Months	5.2
All Months, Internal	1.9
All Months, Internal, # recipients < 20	1.4
October	2.1
October, Internal	1.4
October, Internal # recipients < 20	1.1
November	2.5
November, Internal	1.5
November, Internal, # recipients < 20	1.1
December	2.4
December, Internal	1.4
December, Internal, # recipients < 20	1.1

Table 13: Database Query Times in Seconds

4.2.2 Data Usefulness Test Results.

All 12 generated sociograms were imported by UCINet successfully and the Freeman Degree, Bonacich Power, and k-core social network metrics were taken. The entire network and certain ego networks were also visualized in NetDraw. Thus, the

data created by the system is considered usable. Screenshots of UCINet outputs for those metrics are shown in Figures 33-35.

The social network data can be visualized by NetDraw as shown in Figures 36-42. Figure 36 displays the sociogram corresponding to social network data in December 2004. While complete, this visualization provides little information and must be filtered. Figure 37 shows the egonet of Actor 59 in NetDraw and Figure 38 shows a more readable subset of the December 2004 network. Figures 39-42 showcase different visualizations available in Netdraw for the egonet of Actor 1141.

4.3 Summary

The results show that useful social network data can be created by the system developed in a timely manner. The total time to convert three months of SMTP logs into social network data is about 80 minutes and is well within a day. The data created by this system is readable by UCINet and can be visualized by NetDraw.

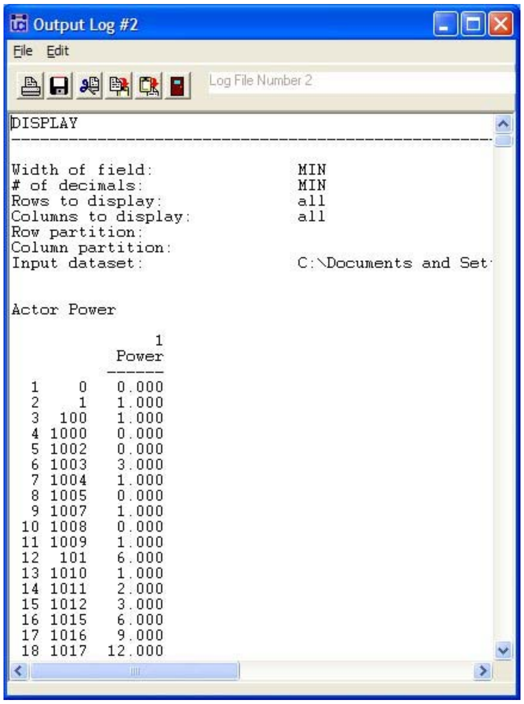

Figure 33: Bonacich Power Metric

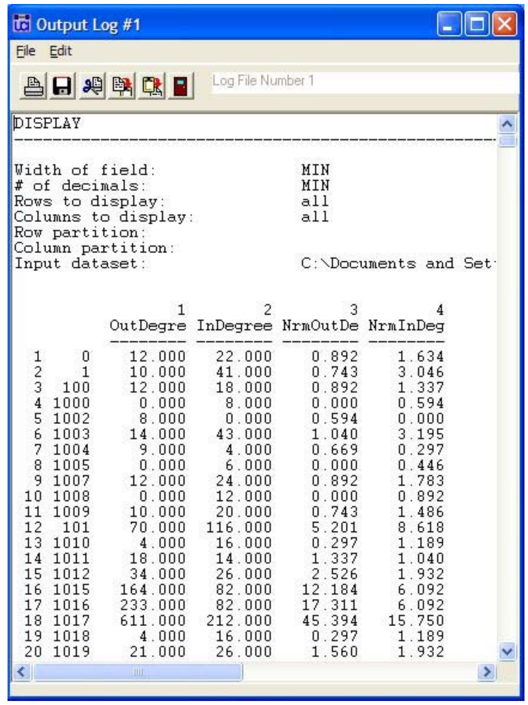

Figure 34: UCINet Freeman Degree Metric

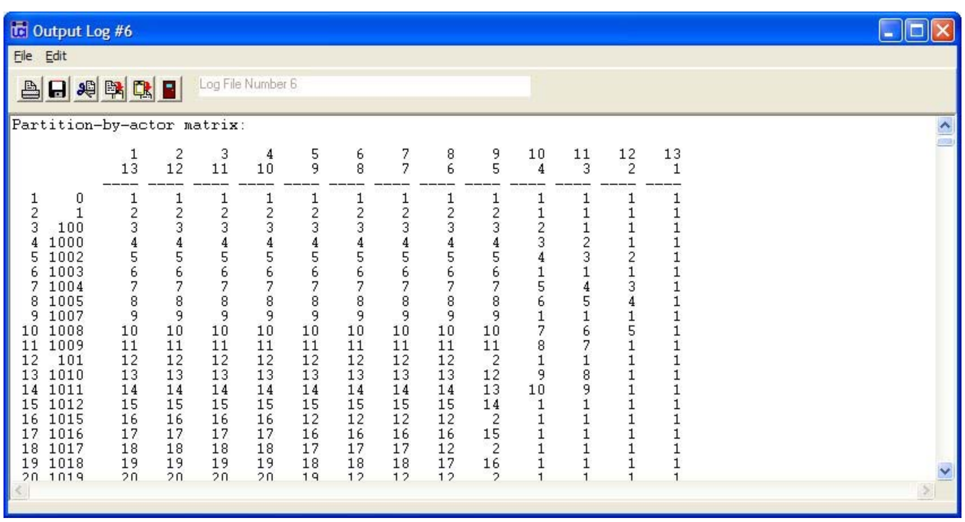

Figure 35: UCINet Output for k-core Metric

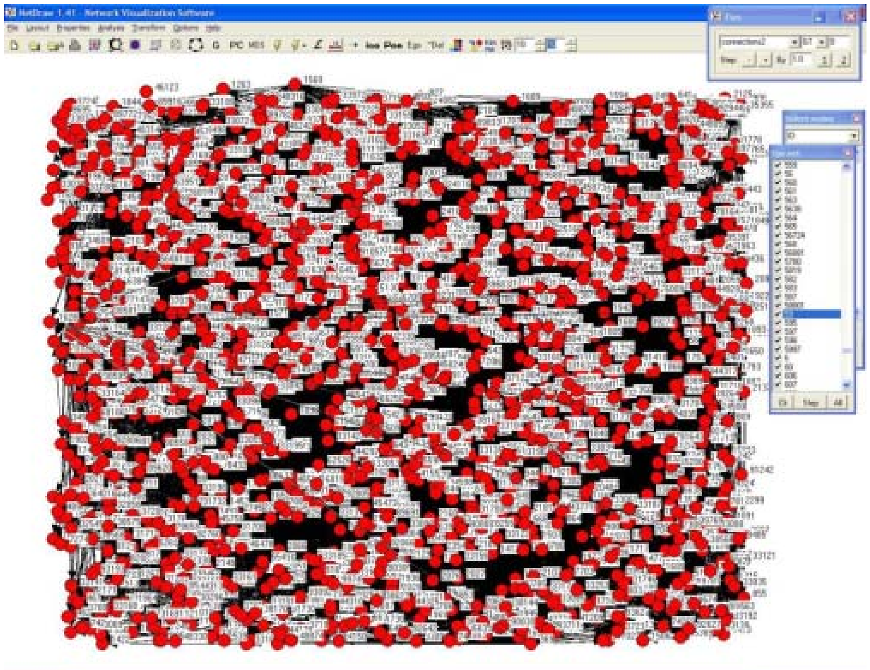

Figure 36: NetDraw Visualization of Social Network Data for December 2004

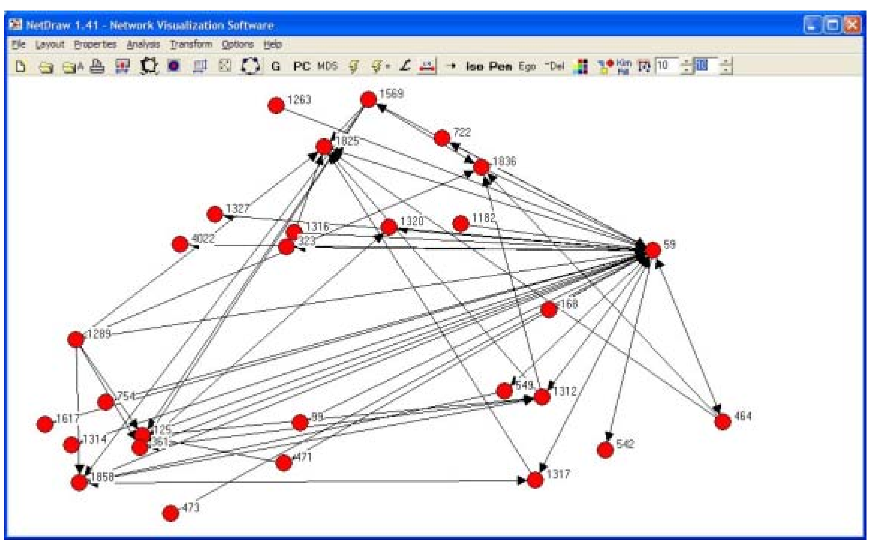

Figure 37: Egonet of Actor with UID 59

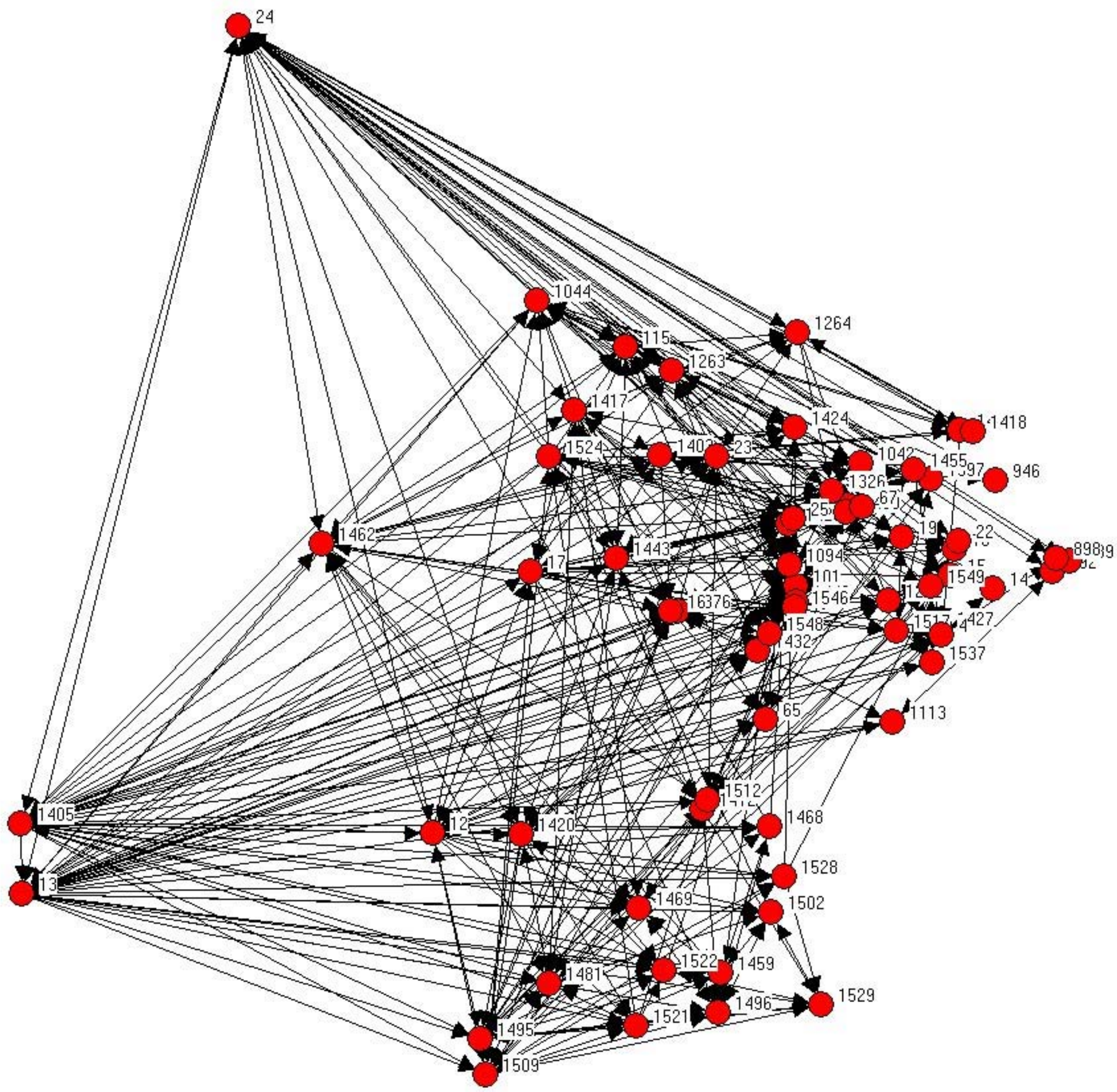

Figure 38: Subset of the December 2004 Sociogram

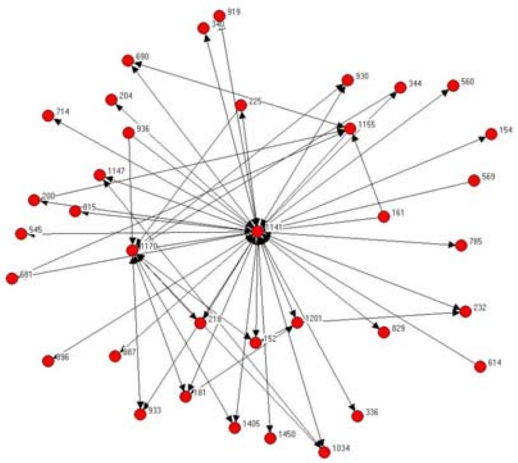

Figure 39: Spring-Embedding Visualization

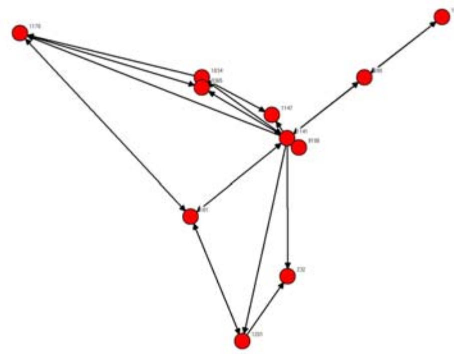

Figure 40: Gower Visualization

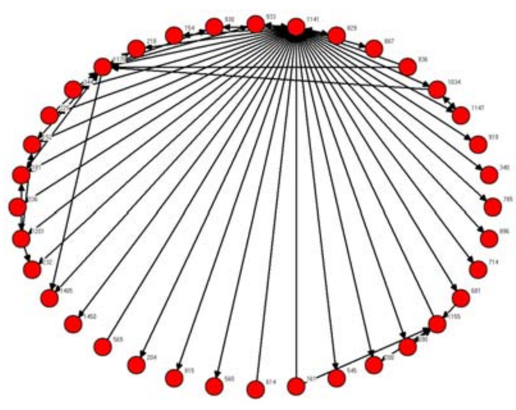

Figure 41: Circular Visualization

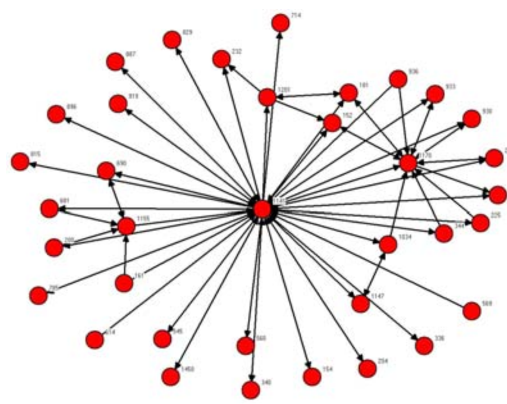

Figure 42: Multi-Dimensional Scaling Visualization

V. Conclusion and Recommendations

This chapter reviews the research objectives, discusses the immediate and long-term impact the proof of concept tool, and proposes goals and follow on work for this research.

5.1 Research Objectives

The goal of this research is to efficiently generate social network data from computer-mediated communication logs. The timeliness of execution and usability of data created by the system were tested and as shown in the results of the experiment, the developed system is able to generate social network data corresponding to three months of SMTP logs in about 80 minutes. Additionally, the results showed that the social network data created by the system is readable, analyzable, and visualizable by standard social network analysis programs.

5.2 Impact

The findings of this research provide a great benefit to the field of social network analysis and in turn impact the ability to monitor the behavior of employees and mitigate the insider threat in the long run.

5.2.1 Immediate Impact.

The immediate impact of the software developed for this research is that social network data of large groups is now available for social network analysis. This data can hopefully provide some breakthroughs in the field of social network analysis in terms of analyzing the short-term changes in social behavior and structure.

5.2.2 Long-Term Impact.

The long-term impact of this research is that longitudinal social network data studies are now much easier to carry out as costly and time-consuming surveys are no longer necessary. SMTP logs are available as long as there is a need for email, and

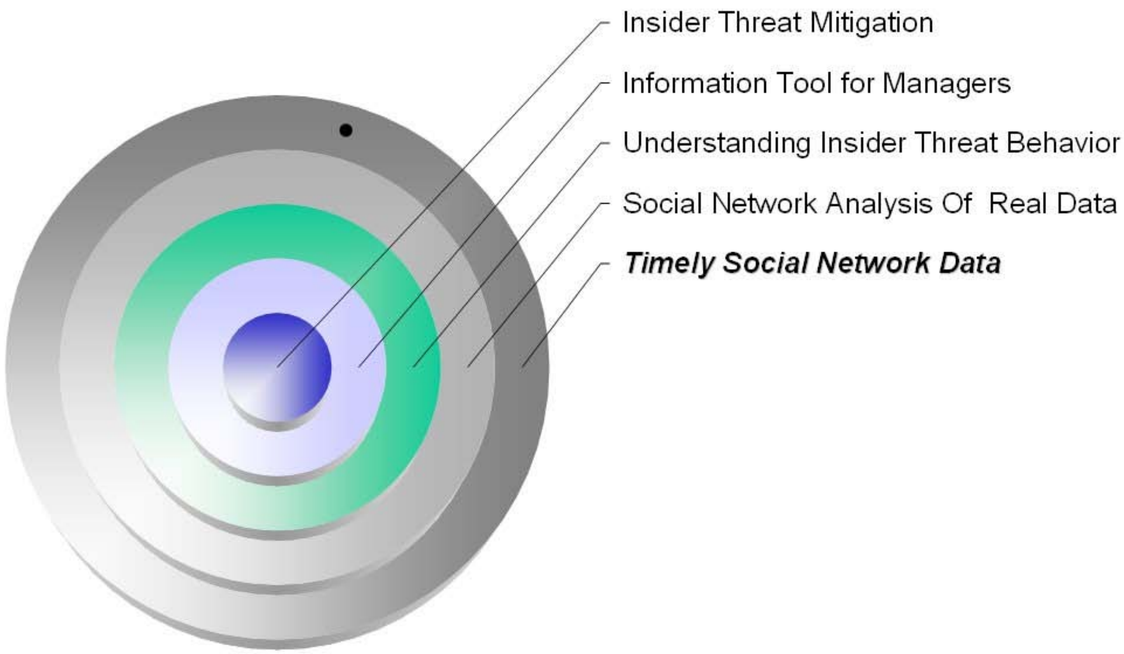

Figure 43: Future Targets for Using SNA to Mitigate the Insider Threat

this research proved that social network data is extractable from a month of SMTP data in under half an hour.

In terms of mitigating the insider threat, the long-term impact of this research is dependent on the work of social network analysts. The future research of social network analysts should focus on finding a way to characterize normal and abnormal email behavior or deriving a behavior-based characterization for the insider threat. These findings can be leveraged to generate an information providing tool for managers.

5.3 Future Goals

Future goals that ultimately result in mitigating the insider threat through social network analysis are shown in Figure 43. Creating timely social network data is the necessary first step that makes the other steps possible. Some future research necessary to reach this goal are outlined in this section.

5.3.1 Characterizing Behavior.

The social network data generated by this research should be statistically analyzed to determine if there are any trends or characterizations of user behavior. In terms of mitigating the insider threat, social network analysis may focus on characterizing the insider threat given representative sets of data. There are many different methods and metrics of social network analysis and hopefully, some are able to provide useful information for characterization. As stated earlier, a characterization of user behavior can be used to create a tool that will provide additional information to managers about their subordinates.

5.3.2 Gathering More Live Data.

At the time that this research was conducted, there was no longitudinal data of an insider developing or attacking. This made the research very difficult to validate because there was nothing to attempt to detect. The insider threat is unique in this aspect. Hopefully, organizations that have live data of an insider threat in action will be willing to share it after sanitizing it, or be willing to share the social network data generated. For example, the Enron email corpus could be used for this purpose [6].

5.3.3 Determining the Best Parameters.

Future research should try to determine the best parameters to use in the generation of social network data. These parameters should be chosen so that the resulting data reflects actual social structure and behavior. These parameters include:

- Date range
- Date range selection
- Min/max thresholds for number of recipients
- Min/max thresholds for number of interactions

Knowing these parameters will help create more meaningful social network data and in turn provide better results from analysis.

5.3.4 Expansion to Other CMC Records.

As implemented, the system only works on records of email communication. However, there are many different types of CMCs, with email being one of the most well-known and accessible. Other sources of CMC records are phone logs, instant messenging records, and web-access logs.

The more data available for research, the better. The data must be used correctly, but there is no harm in having more. More data interpreted correctly gives a greater understanding of the users who create them.

5.3.5 Tool Improvements.

Possible improvements to the tool created in this research are listed in this subsection.

5.3.5.1 UIDList Database.

Instead of using a user identification text file, use a database. A system with 512 MB of RAM ran out of memory while processing the tens of thousands of different email addresses in the sanitization process.

5.3.5.2 Parallelization.

The `ProxyListToUID` program is parallelizable and distributed computing methods can be used to exploit this property. This would greatly speed up the process of converting SMTP logs to social network data.

5.3.5.3 Data Sanitization.

The SMTPLogSanitizer in its current state only partially sanitizes the data, sanitizing the username and leaving the domain name. For example `jasonyee@af.mil` becomes `<uniqueIDnumber>@af.mil`. While this is essentially sufficient for the privacy of the users, for an even more anonymous solution, no evidence of domain name should be present either. Doing so would require that there is a separate list of which users are internal to the system.

5.3.5.4 Unification, GUI.

As currently implemented, the system is a process consisting of independent parts and no GUI. To enhance this system and increase the ease of use, the components can be combined together in one program with a GUI.

5.4 Summary

This research provided a means of efficiently generating social network data from records of computer mediated communications. This tool has an immediate impact upon the field of social network analysis and lays the foundation for a tool that can potentially mitigate the insider threat.

Bibliography

1. Adamic, Lada and Eytan Adar. *How to Search a Social Network*. Technical report, HP Labs, October 2004.

2. Anchor, Kevin, Jesse Zydallis, Gregg Gunsch, and Gary Lamont. *Extending the Computer Defense Immune System: Network Intrusion Detection with a Multiobjective Evolutionary Programming Approach*. Technical report, Air Force Institute of Technology, 2002.

3. Batagelj, Vladimir and Andrej Mrvar. "Pajek - Program for Large Network Analysis". *Connections*, 21(2), 1998.

4. Borgatti, S.P., M.G Everett, and L.C. Freeman. *Ucinet for Windows: Software for Social Network Analysis*. Analytic Technologies, Harvard, 2002.

5. Coe, Kathy. *Behind the Firewall - The Insider Threat*. Technical report, Symantec, March 2004. Http://www.eweek.com/article2/0,1759,1543223,00.asp.

6. Cohen, William. "Enron Email Dataset", 2004. Http://www-2.cs.cmu.edu/ enron.

7. Cross, Rob. "RobCross.org", 2005. Http://www.robcross.org/.

8. "Defense In Depth, NSA Security Configuration Guides", 2004. Http://www.nsa.gov/snac/downloads_docs.cfm?MenuID=scg10.3.1.

9. Freeman, Linton. "Visualizing Social Networks". *Journal of Social Structure*, 1(1), 2000.

10. Garton, Laura, Caroline Haythornthwaite, and Barry Wellman. "Studying Online Social Networks". *Journal of Computer-Mediated Communication*, 3(1), 1997. Http://www.ascusc.org/jcmc/vol3/issue1/garton.html.

11. Hanneman, Robert. *Introduction to Social Network Methods*. Department of Sociology, University of California, Riverside, 2001.

12. Hayden, Michael. "The Insider Threat to U.S. Government Information Systems". *NSTISSAM INFOSEC*, 1999. Http://www.nstissc.gov/Assets/pdf/NSTISSAM_INFOSEC1-99.pdf.

13. Jones, Anita and et. al. *Summary of Discussions at a Planning Meeting on Cyber-Security and the Insider Threat to Classified Information*. Technical report, National Research Council, November 2001. Http://www7.nationalacademies.org/cstb/whitepaper_insiderthreat.html.

14. Krackhardt, David. "Krackplot", 2003. Http://www.andrew.cmu.edu/user/krack/krackplot/krackindex.html.

15. Krebs, Vladis. "The Social Life of Routers". *The Internet Protocol Journal*, 3(4), December 2000.

16. Krebs, Vladis. "Uncloaking Terrorist Networks", 2001. Http://www.firstmonday.org/issues/issue7_4/krebs/.

17. Krebs, Vladis. "How to do Social Network Analysis", 2005. Http://www.orgnet.com/sna.html.

18. McCarty, Chris. "Structure in Personal Networks". *Journal of Social Structure*, 3(1), 2002. Http://www.cmu.edu/joss/content/articles/volume3/McCarty.html.

19. Microsoft. "Log Formats", 2005. Http://www.microsoft.com/resources /documentation/WindowsServ/2003/standard/proddocs/en-us/Default.asp? url=/resources/documentation/WindowsServ/2003/standard/proddocs/en-us/smtp_monitoring_log_formats.asp.

20. Microsoft. "Trustworthy Computing", 2005. Http://www.microsoft.com/mscorp/twc/default.mspx.

21. Morville, Peter. "Social Network Analysis". *SemanticStudios*, 2002. Http://semanticstudios.com/publications/semantics/000006.php.

22. Mutton, Paul. "PieSpy Social Network Bot", 2003. Http://www.jibble.org/piespy/.

23. "Presidential Decision Directive 63, The Clinton Administration's Policy on Critical Infrastructure Protection", 1998. Http://www.usdoj.gov/criminal/cybercrime/white_pr.htm.

24. RAND, Robert Anderson, Thomas Bozek, Tom Longstaff, Wayne Meizler, Michael Skroch, and Ken Van Wyk. "Research on Mitigating the Insider Threat to Information Systems". *Proceedings of a Workshop Held August, 2000*, 111. 2000. Http://www.rand.org/publications/CF/CF163/.

25. Randazzo, Marisa, Dawn Cappelli, and Et Al. *Insider Threat Study: Illicit Cyber Activity in the Banking and Finance Sector*. Technical report, Software Engineering Institute, CMU, August 2004. Http://www.cert.org/archive/pdf/bankfin040820.pdf.

26. Richardson, David and Brent Presley. "MAGE version 6.02", 2002. Mage@kinemage.biochem.duke.edu.

27. Richardson, Robert and Lawrence Gordon. *2004 CSI/FBI Computer Crime and Security Survey*. Technical report, Computer Security Institute, 2004.

28. Shaw, Eric, Jerrold Post, and Kevin Ruby. *Managing the Threat From Within*. Technical report, InfoSecurityMag.techtarget.com, 2000. Http://infosecuritymag.techtarget.com/articles/july00/features2.shtml.

29. Shaw, Eric, Jerrold Post, and Kevin Ruby. “Inside The Mind of the Insider”. *Security Management Online*, 2002. Http://www.securitymanagement.com/library/000762.html.

30. Shaw, Eric, Keven Ruby, and Jerrold Post. “The Insider Threat To Information Systems”. *Security Awareness Bulletin*, 2(98), 1997. Http://rf-web.tamu.edu/security/secguide/Treason/Infosys.htm.

31. Snort.org. “Snort - The Open Source Network Intrusion Detection System”, accessed 2005. Http://www.snort.org/.

32. Spitzner, Lance. “Honeypots: Catching the Insider Threat”, 2003. Http://www.acsac.org/2003/papers/spitzner.pdf.

33. Symantec.com. “Symantec Decoy Server”, 2005. Http://enterprisesecurity.symantec.com/products/products.cfm?ProductID=157.

34. of Virginia Network Roundtable, University. “Online Survey”, 2005. Https://webapp.comm.virginia.edu/SnaPortal/Default.aspx?tabid=34.

35. Wasserman, Stanley and Katherine Faust. *Social Network Analysis, Methods and Applications*. Cambridge University Press, Cambridge, UK, 1994.

36. Zhen, Jian. “The war on leaked intellectual property”. *ComputerWorld.com*, 2005. Http://www.computerworld.com/securitytopics/security/story/0,10801,98724,00.html.

Vita

Second Lieutenant Jason Wei Sung Yee was born and raised in Honolulu. Jason graduated from UC Irvine with a degree in Information and Computer Science with a specialization in Networks and Distributed Systems in 2003. He received his commission through AFROTC at Loyola Marymount University and placed in career field of Communication and Information. AFIT is Jason's first assignment and his follow on assignment at Hickam AFB will bring him back home.

While at Wright-Patterson AFB, Jason played on AFIT's Softball, Basketball, Soccer, and Bowling intramural sports teams. He was the team captain the co-ed softball team that won the WPAFB Base Championship. He also volunteered with the Airmen Against Drunk Driving (AADD) and developed the freeware Enqueueee program for Winamp.

Jason is scheduled to graduate from AFIT in March 2005 with a Masters Degree in Computer Science with an emphasis on Information Assurance. He is a member of the Golden Key, Phi Beta Kappa, and Tau Beta Pi Honor Societies.

Permanent address: jasonwsyee@gmail.com

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*
21–03–2005

2. REPORT TYPE
Master's Thesis

3. DATES COVERED *(From — To)*
Sept 2003 — Mar 2005

4. TITLE AND SUBTITLE
Efficient Generation of
Social Network Data from
Computer-Mediated Communication Logs

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Jason Wei Sung Yee, 2d Lt, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology
Graduate School of Engineering and Management
2950 Hobson Way, Bldg 641
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GCS/ENG/05-19

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Mr. Will Janssen
National Security Agency
9800 Savage Road
Suite 6773
Ft. Meade MD, 20755
(410) 854-4747 - wjanssen@nsa.gov

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approval for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The insider threat poses a significant risk to any network or information system. A general definition of the insider threat is an authorized user performing unauthorized actions, a broad definition with no specifications on severity or action. While limited research has been able to classify and detect insider threats, it is generally understood that insider attacks are planned, and that there is a time period in which the organization's leadership can intervene and prevent the attack. Previous studies have shown that the person's behavior will generally change, and it is possible that social network analysis could be used to observe those changes.
Unfortunately, generation of social network data can be a time consuming and manually intensive process. This research discusses the automatic generation of such data from computer mediated communication records. Using the tools developed in this research, raw social network data can be gathered from communication logs quickly and cheaply. Ideas on further analysis of this data for insider threat mitigation are then presented.

15. SUBJECT TERMS
social network analysis, insider threat, information assurance, data mining

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
U	U	U	UU	83	Dr. Robert Mills
					19b. TELEPHONE NUMBER *(include area code)* (937) 255–3636, ext 4527 - Robert.Mills@afit.edu